문제 01

탐욕 알고리즘의 일반적인 진행 방식으로 가장 적절한 것은?

1. 크기가 같은 부분 문제를 매번 새로 계산한다
2. 각 단계의 선택을 다음 단계마다 반드시 철회한다
3. 최적 부분 구조가 있으면 기준과 무관하게 최적이다
4. 현재 선택을 확정하기 전에 남은 모든 완성 해를 비교한다
5. 현재 기준에서 최선인 선택을 확정하며 진행한다

문제 01 정답과 해설

탐욕 알고리즘의 일반적인 진행 방식으로 가장 적절한 것은?

5번 현재 기준에서 최선인 선택을 확정하며 진행한다

해설

탐욕법은 국소적인 선택을 확정한다. 이 선택이 전체 최적해로 이어지는지는 별도의 증명이 필요하다.

문제 02

최적 부분 구조만 확인한 문제에 대해 내릴 수 있는 결론은?

1 탐욕 선택의 정당성은 추가로 확인해야 한다

2 최적해가 유일할 때만 탐욕법을 쓸 수 있다

3 부분 문제가 겹치지 않으면 탐욕법이 자동으로 성립한다

4 최적 부분 구조만으로 임의의 탐욕 기준이 증명된다

5 국소 최적해가 존재하면 전체 최적해와 항상 같다

문제 02 정답과 해설

최적 부분 구조만 확인한 문제에 대해 내릴 수 있는 결론은?

1번 탐욕 선택의 정당성은 추가로 확인해야 한다

해설

최적 부분 구조와 탐욕 선택 속성은 서로 다른 조건이다. 0/1 배낭은 최적 부분 구조가 있지만 가치 밀도 탐욕은 실패한다.

문제 03

탐욕 선택을 정당화하는 교환 논증의 핵심은?

1. 탐욕 선택의 실행 시간이 짧으면 목적값도 좋다고 본다
2. 최적해의 일부를 탐욕 선택으로 바꿔도 목적값이 나빠지지 않음을 보인다
3. 가장 작은 예제에서 탐욕과 최적해가 같은지만 확인한다
4. 탐욕 선택을 포함하는 실행 가능한 해 하나만 보인다
5. 탐욕 선택이 포함되지 않은 최적해는 없다고 증명 없이 가정한다

문제 03 정답과 해설

탐욕 선택을 정당화하는 교환 논증의 핵심은?

2번 최적해의 일부를 탐욕 선택으로 바꿔도 목적값이 나빠지지 않음을 보인다

해설

최적해를 탐욕 선택을 포함하는 최적해로 바꿀 수 있으면 첫 선택의 안전성을 설명할 수 있다.

문제 04

탐욕 알고리즘이 항상 최적이라는 주장을 반박하기에 충분한 것은?

1. 조건을 위반한 입력에서 오답을 제시한다
2. 조건을 만족하는 입력 하나에서 더 나은 해를 제시한다
3. 같은 목적값을 갖는 다른 해를 제시한다
4. 탐욕 결과가 느린 입력을 하나 찾는다
5. 작은 입력 모두에서 성공했음을 보인다

문제 04 정답과 해설

탐욕 알고리즘이 항상 최적이라는 주장을 반박하기에 충분한 것은?

2번 조건을 만족하는 입력 하나에서 더 나은 해를 제시한다

해설

보편 명제는 유효한 반례 하나로 깨진다. 단순히 다른 해가 아니라 더 좋은 목적값을 보여야 한다.

문제 05

큰 동전이 바로 다음 작은 동전의 정수배인 양의 정수 화폐 체계에 1원이 포함된다. 각 동전을 무제한 사용할 때 최소 동전 수를 구하는 방법은?

1 동전의 종류 수를 최소화한다

2 동전 종류마다 반드시 하나씩 선택한다

3 작은 동전부터 가능한 만큼 선택한다

4 큰 동전부터 가능한 만큼 선택한다

5 큰 동전은 항상 하나만 사용한다

문제 05 정답과 해설

큰 동전이 바로 다음 작은 동전의 정수배인 양의 정수 화폐 체계에 1원이 포함된다. 각 동전을 무제한 사용할 때 최소 동전 수를 구하는 방법은?

4번 큰 동전부터 가능한 만큼 선택한다

해설

작은 동전 여러 개를 같은 금액의 큰 동전 하나로 교환하면 개수가 줄어든다. 정수배 관계가 이 교환을 뒷받침한다.

문제 06

동전 {1,3,4}로 6을 만드는 최소 동전 수는?

1. 1개
2. 4개
3. 6개
4. 3개
5. 2개

문제 06 정답과 해설

동전 {1,3,4}로 6을 만드는 최소 동전 수는?

5번 2개

해설

큰 동전 우선은 4+1+1로 3개지만 최적해는 3+3으로 2개다.

문제 07

동전 {1,5,10,50}을 무제한 사용할 때 큰 동전 우선으로 28을 만들고 남는 금액은?

1 0

2 5

3 8

4 1

5 3

문제 07 정답과 해설

동전 {1,5,10,50}을 무제한 사용할 때 큰 동전 우선으로 28을 만들고 남는 금액은?

1번 0

해설

10원 2개, 5원 1개, 1원 3개를 사용한다. 1원이 있으므로 정확히 만들 수 있다.

문제 08

동전 종류가 m개이고 이미 내림차순 정렬되어 있다. 나눗셈과 나머지로 개수를 구하는 탐욕 루프의 시간은? 금액 연산은 상수 시간이다.

1 Θ(log m)

2 Θ(1), m과 무관

3 Θ(m)

4 Θ(금액)

5 Θ(m^2)

문제 08 정답과 해설

동전 종류가 m개이고 이미 내림차순 정렬되어 있다. 나눗셈과 나머지로 개수를 구하는 탐욕 루프의 시간은? 금액 연산은 상수 시간이다.

3번 Θ(m)

해설

각 종류를 한 번씩 처리한다. 동전을 하나씩 빼는 구현과 구별해야 한다.

문제 09

활동 선택 문제에서 [시작, 종료) 구간을 사용한다. 직전 활동이 시각 5에 끝났을 때 다음 활동의 허용 조건은?

1. 시작 시각 ≥ 4
2. 시작 시각 ≥ 5
3. 시작 시각 ≤ 5
4. 시작 시각 > 5
5. 종료 시각 ≥ 5이면 시작은 무관하다

문제 09 정답과 해설

활동 선택 문제에서 [시작, 종료) 구간을 사용한다. 직전 활동이 시각 5에 끝났을 때 다음 활동의 허용 조건은?

2번 시작 시각 ≥ 5

해설

반열린 구간에서는 끝점이 닿는 두 활동이 겹치지 않는다.

문제 10

모든 활동의 가치가 같고 겹치지 않는 활동 수를 최대화할 때 안전한 탐욕 기준은?

1. 종료 시각이 빠른 순서
2. 종료 시각이 늦은 순서
3. 시작 시각이 빠른 순서
4. 겹치는 활동이 적은 순서
5. 기간이 짧은 순서

문제 10 정답과 해설

모든 활동의 가치가 같고 겹치지 않는 활동 수를 최대화할 때 안전한 탐욕 기준은?

1번 종료 시각이 빠른 순서

해설

가장 빨리 끝나는 활동을 고르면 이후 활동을 위한 시간을 충분히 남긴다.

문제 11

활동을 종료 시각으로 이미 정렬했다. 선택 여부를 결정하기 위해 유지할 핵심 상태는?

1. 가장 짧은 활동의 길이
2. 마지막으로 검사한 활동의 종료 시각
3. 선택한 활동들의 길이 합
4. 처음 선택한 활동의 종료 시각
5. 마지막으로 선택한 활동의 종료 시각

문제 11 정답과 해설

활동을 종료 시각으로 이미 정렬했다. 선택 여부를 결정하기 위해 유지할 핵심 상태는?

5번 마지막으로 선택한 활동의 종료 시각

해설

새 활동의 시작이 마지막 선택의 종료 이상이면 선택한다. 거절한 활동으로 종료 시각을 갱신하면 안 된다.

문제 12

활동마다 보상이 다를 때 종료 시각 우선으로 활동 수를 최대화한 결과는?

1 보상값이 정수이면 같은 탐욕 증명이 성립한다

2 보상이 양수이면 활동 수와 보상 최대화가 같다

3 총보상도 반드시 최대가 된다

4 총보상을 최대화하지 못할 수 있다

5 종료 시각 동률만 보상순으로 풀면 항상 최적이다

문제 12 정답과 해설

활동마다 보상이 다를 때 종료 시각 우선으로 활동 수를 최대화한 결과는?

4번 총보상을 최대화하지 못할 수 있다

해설

목적 함수가 개수에서 보상 합으로 바뀌면 같은 탐욕 증명이 적용되지 않는다.

문제 13

분할 가능 배낭에서 가치와 무게가 양수일 때 먼저 담을 기준은?

1. 단위 무게당 가치가 큰 순서
2. 가치와 무게의 차이가 큰 순서
3. 총가치가 큰 순서
4. 무게가 작은 순서
5. 단위 가치당 무게가 큰 순서

문제 13 정답과 해설

분할 가능 배낭에서 가치와 무게가 양수일 때 먼저 담을 기준은?

1번 단위 무게당 가치가 큰 순서

해설

높은 밀도의 물건을 낮은 밀도의 물건으로 바꾸면 같은 무게에서 가치가 줄어든다.

문제 14

분할 가능 배낭에서 마지막 물건을 전부 담을 수 없을 때 올바른 처리는?

1. 용량을 초과해 담은 뒤 총가치만 비례 축소한다
2. 가치는 전부 더하고 무게만 남은 용량으로 센다
3. 남은 용량만큼의 비율을 담는다
4. 물건을 버리고 뒤의 모든 물건도 검사하지 않는다
5. 가치와 무게를 각각 서로 다른 비율로 줄인다

문제 14 정답과 해설

분할 가능 배낭에서 마지막 물건을 전부 담을 수 없을 때 올바른 처리는?

3번 남은 용량만큼의 비율을 담는다

해설

무게와 가치를 같은 비율로 취해야 한다.

문제 15

0/1 배낭과 분할 가능 배낭의 차이로 옳은 것은?

1. 양의 가치를 가지면 두 문제의 최적값은 항상 같다
2. 0/1 배낭은 물건의 일부만 선택할 수 없다
3. 0/1 배낭은 한 종류의 물건만 선택할 수 있다
4. 분할 가능 배낭은 한 물건을 무제한 복제할 수 있다
5. 분할 가능 배낭에서는 무게 제한이 사라진다

문제 15 정답과 해설

0/1 배낭과 분할 가능 배낭의 차이로 옳은 것은?

2번 0/1 배낭은 물건의 일부만 선택할 수 없다

해설

분할 허용 여부가 해 집합과 탐욕 선택의 정당성을 바꾼다.

문제 16

밀도가 같은 물건의 순서가 바뀌면 분할 가능 배낭의 최대 가치에 어떤 영향이 있는가? 무게와 가치는 양수다.

1. 최대 가치는 변하지 않는다
2. 밀도 동률이 있으면 탐욕법의 정당성이 깨진다
3. 무게가 작은 물건을 먼저 골라야 가치가 더 크다
4. 총가치가 큰 물건을 먼저 골라야 가치가 더 크다
5. 동률 물건이 두 개 이상이면 최적값을 정할 수 없다

문제 16 정답과 해설

밀도가 같은 물건의 순서가 바뀌면 분할 가능 배낭의 최대 가치에 어떤 영향이 있는가? 무게와 가치는 양수다.

1번 최대 가치는 변하지 않는다

해설

같은 무게를 담았을 때 얻는 가치가 같으므로 밀도가 같은 물건끼리의 교환은 목적값을 바꾸지 않는다.

문제 17

연결된 무방향 그래프에서 정점이 V개인 신장 트리의 간선 수는?

1. 2V
2. V
3. V+1
4. V−1
5. V(V−1)/2

문제 17 정답과 해설

연결된 무방향 그래프에서 정점이 V개인 신장 트리의 간선 수는?

4번 V−1

해설

트리는 연결되어 있고 사이클이 없으며 V−1개의 간선을 가진다.

문제 18

Kruskal 알고리즘이 간선을 검사하는 순서는?

1. 가중치가 큰 순서
2. 간선 양 끝의 차수 합이 작은 순서
3. 가중치가 작은 순서
4. 출발점에서의 잠정 거리가 작은 순서
5. 정점 번호가 작은 간선 순서

문제 18 정답과 해설

Kruskal 알고리즘이 간선을 검사하는 순서는?

3번 가중치가 작은 순서

해설

가벼운 간선부터 보되 사이클을 만드는 간선은 제외한다.

문제 19

Kruskal에서 간선 (u,v)를 선택하는 조건은?

1. 적어도 한쪽 정점이 아직 어떤 간선에도 연결되지 않았다
2. 선택 후 모든 정점의 차수가 2 이하가 된다
3. u와 v가 현재 서로 다른 연결 요소에 있다
4. 가중치가 지금까지의 비용 합보다 작다
5. 두 정점이 이미 같은 대표를 갖는다

문제 19 정답과 해설

Kruskal에서 간선 (u,v)를 선택하는 조건은?

3번 u와 v가 현재 서로 다른 연결 요소에 있다

해설

서로 다른 트리를 연결할 때만 간선을 더하면 사이클을 만들지 않는다.

문제 20

분리 집합 자료구조에서 find(x)가 나타내는 것은?

1. x에서 모든 정점까지 거리의 최솟값
2. x가 속한 집합의 대표 원소
3. x의 부모까지 간선 가중치
4. x가 처음 연결된 이웃 정점
5. x가 속한 집합의 원소 수

문제 20 정답과 해설

분리 집합 자료구조에서 find(x)가 나타내는 것은?

2번 x가 속한 집합의 대표 원소

해설

같은 대표를 갖는 정점은 같은 연결 요소에 있다.

문제 21

연결되지 않은 무방향 그래프에 Kruskal을 적용하면 얻는 것은?

1. 각 요소에서 가장 가벼운 간선 하나씩만 고른 집합
2. 각 연결 요소의 최소 신장 트리로 이루어진 숲
3. 모든 정점을 연결하는 단일 트리
4. 모든 연결 요소를 한 번씩 잇는 순회
5. 출발점에서 도달하는 정점만 포함한 최단 경로 트리

문제 21 정답과 해설

연결되지 않은 무방향 그래프에 Kruskal을 적용하면 얻는 것은?

2번 각 연결 요소의 최소 신장 트리로 이루어진 숲

해설

연결 요소 사이에 간선이 없으므로 전체 신장 트리는 존재하지 않는다.

문제 22

Kruskal이 음수 가중치 간선을 처리할 때 옳은 설명은?

1 음수 간선이 있으면 간선 정렬 순서를 반대로 한다

2 음수 간선이 있으면 MST가 정의되지 않는다

3 음수 간선을 0으로 바꾸면 MST 비용도 보존된다

4 음수 간선도 정렬하여 정상적으로 처리할 수 있다

5 음수 간선은 사이클 여부와 무관하게 모두 선택한다

문제 22 정답과 해설

Kruskal이 음수 가중치 간선을 처리할 때 옳은 설명은?

4번 음수 간선도 정렬하여 정상적으로 처리할 수 있다

해설

MST의 탐욕 정당성은 간선 가중치의 비음수 조건을 요구하지 않는다.

문제 23

연결된 그래프의 모든 간선 가중치가 서로 다르면 MST에 대해 성립하는 것은?

1. MST의 간선들이 모든 정점 쌍의 최단 경로를 보존한다
2. MST가 유일하다
3. 가장 가벼운 V−1개 간선을 검사 없이 선택해도 된다
4. MST의 차수가 모든 정점에서 같아진다
5. MST의 총비용이 반드시 양수다

문제 23 정답과 해설

연결된 그래프의 모든 간선 가중치가 서로 다르면 MST에 대해 성립하는 것은?

2번 MST가 유일하다

해설

서로 다른 가중치는 MST 유일성의 충분조건이다. 같은 가중치가 있다고 반드시 여러 MST가 생기는 것은 아니다.

문제 24

간선 정렬이 필요한 일반적인 Kruskal 구현의 시간 차수를 지배하는 항은?

1. Θ(log E)
2. Θ(V log V), E와 무관
3. Θ(E log E)
4. Θ(E)
5. Θ(E^2)

문제 24 정답과 해설

간선 정렬이 필요한 일반적인 Kruskal 구현의 시간 차수를 지배하는 항은?

3번 Θ(E log E)

해설

분리 집합 연산보다 E개 간선의 정렬이 보통 지배적이다.

문제 25

최소 신장 트리와 최단 경로 트리에 대해 옳은 것은?

1. 시작점을 정하면 모든 MST가 같은 최단 경로 트리가 된다
2. 최단 경로 트리는 선택 간선 합도 항상 최소다
3. 최적화하는 대상이 달라 서로 다를 수 있다
4. MST가 시작점에서 모든 최단 거리를 보존한다
5. 간선이 모두 양수이면 두 트리는 반드시 같다

문제 25 정답과 해설

최소 신장 트리와 최단 경로 트리에 대해 옳은 것은?

3번 최적화하는 대상이 달라 서로 다를 수 있다

해설

MST는 선택 간선의 총합을, 최단 경로 트리는 한 출발점에서의 거리를 최적화한다.

문제 26

TSP의 최근접 이웃 휴리스틱에서 다음 도시를 고르는 기준은?

1. 현재 도시에서 가장 가까운 도시, 방문 여부 무관
2. 현재 도시에서 가장 먼 미방문 도시
3. 다음 간선과 귀환 간선의 합이 최소인 도시
4. 출발 도시에서 가장 가까운 미방문 도시
5. 현재 도시에서 가장 가까운 미방문 도시

문제 26 정답과 해설

TSP의 최근접 이웃 휴리스틱에서 다음 도시를 고르는 기준은?

5번 현재 도시에서 가장 가까운 미방문 도시

해설

방문 여부를 확인해야 같은 도시를 반복하지 않는다. 최적 순회 보장은 별도다.

문제 27

TSP 순회 비용을 계산할 때 모든 도시를 방문한 뒤 필요한 것은?

1 귀환 비용을 나머지 간선 평균으로 대체한다

2 귀환 비용을 첫 이동 비용과 같다고 간주한다

3 마지막 도시까지의 경로 비용만 반환한다

4 방문한 간선 중 가장 긴 것을 제거한다

5 마지막 도시에서 출발 도시로 돌아오는 비용을 더한다

문제 27 정답과 해설

TSP 순회 비용을 계산할 때 모든 도시를 방문한 뒤 필요한 것은?

5번 마지막 도시에서 출발 도시로 돌아오는 비용을 더한다

해설

TSP는 출발점으로 돌아오는 순회를 요구한다.

문제 28

최근접 이웃 TSP에 대해 일반적으로 옳은 설명은?

1 모든 간선 가중치가 다르면 항상 최적 순회다

2 출발 도시와 동률 처리에 따라 결과가 달라질 수 있다

3 삼각 부등식이 있으면 항상 최적 순회다

4 가장 가까운 도시를 고르므로 최종 귀환 비용도 최소다

5 출발점이 같으면 동률 규칙과 무관하게 결과가 같다

문제 28 정답과 해설

최근접 이웃 TSP에 대해 일반적으로 옳은 설명은?

2번 출발 도시와 동률 처리에 따라 결과가 달라질 수 있다

해설

가까운 다음 도시를 고르는 판단은 남은 전체 순회의 비용을 고려하지 않는다.

문제 29

Dijkstra의 표준 정당성에 필요한 간선 조건은?

1. 도달 가능한 간선의 가중치가 모두 0 이상이다
2. 음수 사이클만 없으면 음수 간선도 항상 허용된다
3. 가중치는 반드시 엄격히 양수여야 한다
4. 모든 정점까지 간선 수가 같아야 한다
5. 그래프는 반드시 사이클이 없는 방향 그래프여야 한다

문제 29 정답과 해설

Dijkstra의 표준 정당성에 필요한 간선 조건은?

1번 도달 가능한 간선의 가중치가 모두 0 이상이다

해설

비음수 간선이면 가장 짧은 잠정 거리를 확정한 뒤 나중에 더 짧아질 수 없다.

문제 30

간선 (u,v)의 가중치가 w일 때 최단 거리 완화식은?

1 dist[v] = dist[u]+w, 개선 여부 무관

2 dist[v] = max(dist[v], dist[u]+w)

3 dist[v] = min(dist[v], dist[u]−w)

4 dist[v] = min(dist[v], dist[u]+w)

5 dist[v] = min(dist[v], w)

문제 30 정답과 해설

간선 (u,v)의 가중치가 w일 때 최단 거리 완화식은?

4번 dist[v] = min(dist[v], dist[u]+w)

해설

u까지의 경로에 간선 하나를 붙인 후보와 기존 거리를 비교한다.

문제 31

Dijkstra에서 출발점의 초기 거리와 다른 정점의 초기 거리는?

1 모든 정점을 출발점의 차수로 초기화

2 출발점 0, 나머지도 0

3 출발점 ∞, 나머지 0

4 출발점 1, 나머지 ∞

5 출발점 0, 나머지 ∞

문제 31 정답과 해설

Dijkstra에서 출발점의 초기 거리와 다른 정점의 초기 거리는?

5번 출발점 0, 나머지 ∞

해설

출발점까지의 빈 경로 길이는 0이고 나머지는 아직 경로를 발견하지 못했다.

문제 32

최소 힙 Dijkstra에서 꺼낸 거리 d가 현재 dist[u]보다 크면 어떻게 처리하는가?

1. 현재 거리를 더 큰 d로 다시 덮어쓴다
2. 오래된 항목이 있으면 음수 간선이 있다고 결론낸다
3. 그 항목의 거리로만 이웃을 다시 확정한다
4. 오래된 항목이므로 건너뛴다
5. 해당 정점의 모든 남은 힙 항목을 오류로 처리한다

문제 32 정답과 해설

최소 힙 Dijkstra에서 꺼낸 거리 d가 현재 dist[u]보다 크면 어떻게 처리하는가?

4번 오래된 항목이므로 건너뛴다

해설

더 짧은 경로가 이미 발견된 뒤 남은 힙 항목이므로 재처리가 필요 없다.

문제 33

Dijkstra가 끝난 뒤 dist[v]=∞라면 무엇을 뜻하는가?

1. v가 어떤 음수 사이클 위에 있다
2. v까지 비용이 0인 경로가 있다
3. v에 들어오는 간선이 전혀 없다
4. 출발점에서 v에 도달할 수 없다
5. v는 그래프 전체에서 고립된 정점이다

문제 33 정답과 해설

Dijkstra가 끝난 뒤 dist[v]=∞라면 무엇을 뜻하는가?

4번 출발점에서 v에 도달할 수 없다

해설

다른 연결 요소 내부에 간선이 있더라도 출발점에서 도달할 수 없을 수 있다.

문제 34

배열로 미확정 정점 중 최소 거리를 매번 찾는 Dijkstra의 시간 차수는? 인접 행렬을 사용한다.

1. $\Theta(\log V)$
2. $\Theta(V)$
3. $\Theta(V^2)$
4. $\Theta(V^3)$
5. $\Theta(V \log V)$

문제 34 정답과 해설

배열로 미확정 정점 중 최소 거리를 매번 찾는 Dijkstra의 시간 차수는? 인접 행렬을 사용한다.

3번 $\Theta(V^2)$

해설

최대 V번 최소 정점을 찾고 각 단계에서 V개 이웃을 검사한다.

문제 35

가중치가 0인 간선이 있을 때 Dijkstra의 표준 비음수 조건은?

1. 0 간선이 있으면 반드시 음수 사이클이 생긴다
2. 0 간선을 제거해도 모든 최단 거리가 유지된다
3. 만족한다
4. 무방향 그래프일 때만 사용할 수 있다
5. 엄격한 양수 조건을 위반하므로 사용할 수 없다

문제 35 정답과 해설

가중치가 0인 간선이 있을 때 Dijkstra의 표준 비음수 조건은?

3번 만족한다

해설

0은 음수가 아니다. 구현에서 0을 간선 없음 표시로 쓰면 별개의 오류가 생긴다.

문제 36

X 중간고사 출제 제외

모든 작업이 시각 0에 도착하고 한 번에 하나씩 처리한다. 완료 시각 합을 최소화할 정렬은?

1. 누적 완료 시각이 커지도록 하는 순서
2. 처리 시간이 긴 순서
3. 가장 짧은 작업을 마지막에 놓는 순서
4. 처리 시간이 짧은 순서
5. 정렬해도 완료 시각 합은 항상 같다

문제 36 정답과 해설

X 중간고사 출제 제외

모든 작업이 시각 0에 도착하고 한 번에 하나씩 처리한다. 완료 시각 합을 최소화할 정렬은?

4번 처리 시간이 짧은 순서

해설

짧은 작업을 앞에 놓으면 뒤 작업들이 기다리는 시간을 줄일 수 있다.

문제 37

X 중간고사 출제 제외

작업 처리 시간 [a,b]를 [b,a]로 바꿀 때 두 작업의 완료 시각 합 차이(앞 순서−뒤 순서)는? 두 작업 시작은 같은 시각이다.

1 a−b

2 2a+2b

3 a+b

4 0, 항상

5 ab

문제 37 정답과 해설

X 중간고사 출제 제외

작업 처리 시간 [a,b]를 [b,a]로 바꿀 때 두 작업의 완료 시각 합 차이(앞 순서−뒤 순서)는? 두 작업 시작은 같은 시각이다.

1번 a−b

해설

앞 순서는 2a+b, 뒤 순서는 2b+a이다. a≤b이면 앞 순서가 불리하지 않다.

문제 38

X 중간고사 출제 제외

작업 순서가 정해졌을 때 완료 시각 합을 계산하는 방법은?

1 처리 시간의 총합만 반환한다

2 각 작업 시간에 자신의 1 기반 위치를 곱한다

3 완료 시각에서 자기 처리 시간을 모두 빼서 더한다

4 가장 긴 작업의 완료 시각만 반환한다

5 누적 처리 시간을 매번 정답에 더한다

문제 38 정답과 해설

X 중간고사 출제 제외

작업 순서가 정해졌을 때 완료 시각 합을 계산하는 방법은?

5번 누적 처리 시간을 매번 정답에 더한다

해설

뒤 작업의 완료 시각에는 앞 작업의 처리 시간도 포함된다.

문제 39

탐욕 알고리즘의 시간 복잡도를 평가할 때 옳은 것은?

1 최적해를 반환하면 복잡도는 입력 크기와 무관하다

2 정렬 비용과 선택 비용을 항상 곱한다

3 선택 단계가 O(n)이면 정렬 비용은 포함하지 않는다

4 정렬된 입력과 정렬되지 않은 입력의 비용은 항상 같다

5 정렬과 선택 단계의 비용을 모두 포함한다

문제 39 정답과 해설

탐욕 알고리즘의 시간 복잡도를 평가할 때 옳은 것은?

5번 정렬과 선택 단계의 비용을 모두 포함한다

해설

정렬 후 한 번 순회하는 알고리즘은 보통 정렬 비용이 지배한다.

문제 40

최적화 문제의 탐욕 해를 평가하는 올바른 기준은?

1. 제약 만족 여부와 목적값을 함께 확인한다
2. 제약을 일부 위반해도 목적값이 좋으면 최적해다
3. 탐욕 선택이 유일하면 전체 최적성도 자동으로 성립한다
4. 실행 가능한 해이면 목적값과 무관하게 최적해다
5. 작은 예제에서 성공하면 전체 입력의 최적성도 확정된다

문제 40 정답과 해설

최적화 문제의 탐욕 해를 평가하는 올바른 기준은?

1번 제약 만족 여부와 목적값을 함께 확인한다

해설

빠른 결과라도 제약을 위반하거나 더 나은 해가 있으면 최적해가 아니다.

문제 41

동전 {500,100,50,10}으로 1260원을 큰 동전부터 만든다. 종류별 개수와 총개수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 41 정답과 해설

동전 {500,100,50,10}으로 1260원을 큰 동전부터 만든다. 종류별 개수와 총개수를 쓰시오.

[2,2,1,1], 총 6개

해설

1000원을 고른 뒤 260원이 남는다. 100원 2개, 50원 1개, 10원 1개를 더한다.

문제 42

동전 {1,3,4}로 10을 만든다. 큰 동전 우선의 개수와 최적 개수를 각각 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 42 정답과 해설

동전 {1,3,4}로 10을 만든다. 큰 동전 우선의 개수와 최적 개수를 각각 쓰시오.

탐욕 4개, 최적 3개

해설

탐욕은 4+4+1+1이다. 4+3+3은 3개이며 두 동전으로는 최대 8이므로 최적이다.

문제 43

동전 {1,5,10,50}을 무제한 쓴다. 68의 큰 동전 우선 선택 과정을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 43 정답과 해설

동전 {1,5,10,50}을 무제한 쓴다. 68의 큰 동전 우선 선택 과정을 쓰시오.

50+10+5+1+1+1, 6개

해설

종류별 몫은 1,1,1,3이고 마지막 나머지는 0이다.

문제 44

동전 {4,6}으로 8을 만들 때 큰 동전 우선의 남은 금액과 실제 가능한 구성을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 44 정답과 해설

동전 {4,6}으로 8을 만들 때 큰 동전 우선의 남은 금액과 실제 가능한 구성을 쓰시오.

나머지 2, 실제 해 4+4

해설

6을 먼저 고르면 실패하지만 해가 없는 것은 아니다. 1원이 없는 화폐에서 탐욕의 실패와 불가능을 구분한다.

문제 45

활동 A[0,3), B[1,2), C[2,4), D[4,5)를 종료 시각 순으로 탐욕 선택한다. 선택 순서와 개수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 45 정답과 해설

활동 A[0,3), B[1,2), C[2,4), D[4,5)를 종료 시각 순으로 탐욕 선택한다. 선택 순서와 개수를 쓰시오.

B,C,D, 3개

해설

B가 2에 끝나면 C를 선택할 수 있고 C의 끝 4에서 D를 선택할 수 있다.

문제 46

활동 A[0,4), B[0,2), C[2,3), D[3,4)가 있다. A부터 고른 결과와 최적 활동 수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 46 정답과 해설

활동 A[0,4), B[0,2), C[2,3), D[3,4)가 있다. A부터 고른 결과와 최적 활동 수를 쓰시오.

A부터는 1개, 최적은 B,C,D의 3개

해설

가장 먼저 시작하는 활동을 임의로 고르는 규칙은 긴 활동 때문에 여러 짧은 활동을 놓친다.

문제 47

활동 A[0,2), B[1,2), C[2,3), D[3,6)를 (종료, 시작, 이름) 오름차순으로 선택한다.
결과를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 47 정답과 해설

활동 A[0,2), B[1,2), C[2,3), D[3,6)를 (종료, 시작, 이름) 오름차순으로 선택한다. 결과를 쓰시오.

A,C,D

해설

종료가 같은 A와 B 중 시작이 빠른 A를 먼저 검사한다. B는 겹쳐 거절한다.

문제 48

이미 선택한 마지막 종료 시각이 7이다. 다음 활동 [6,8), [7,9), [8,10)을 순서대로 검사한다. 선택되는 활동은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 48 정답과 해설

이미 선택한 마지막 종료 시각이 7이다. 다음 활동 [6,8), [7,9), [8,10)을 순서대로 검사한다.
선택되는 활동은?

[7,9)만 선택

해설

첫 활동은 거절하고 종료 시각 7을 유지한다. 두 번째를 선택하면 종료가 9가 되어 세 번째를 거절한다.

문제 49

활동 A[0,2)의 보상 1, B[2,4)의 보상 1, C[0,4)의 보상 5다. 종료 시각 우선의 보상과 최대 보상은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 49 정답과 해설

활동 A[0,2)의 보상 1, B[2,4)의 보상 1, C[0,4)의 보상 5다. 종료 시각 우선의 보상과 최대 보상은?

탐욕 2, 최대 5

해설

활동 수는 A,B가 많지만 보상은 C 하나가 더 크다.

문제 50

용량 50, 물건 (무게,가치)가 A(10,60), B(20,100), C(30,120)이다. 분할 가능 배낭의 구성과 가치는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 50 정답과 해설

용량 50, 물건 (무게,가치)가 A(10,60), B(20,100), C(30,120)이다. 분할 가능 배낭의 구성과 가치는?

A 전부, B 전부, C의 2/3, 가치 240

해설

밀도는 6,5,4이다. 남은 20만큼 C를 담아 가치 80을 더한다.

문제 51

용량 50, 물건 (무게,가치)가 A(10,60), B(20,100), C(30,120)인 0/1 배낭의 최대 가치는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 51 정답과 해설

용량 50, 물건 (무게,가치)가 A(10,60), B(20,100), C(30,120)인 0/1 배낭의 최대 가치는?

220, B와 C

해설

A부터 밀도순으로 고르면 A+B의 160에 그친다. B+C의 무게는 50, 가치는 220이다.

문제 52

용량 7, 물건 A(무게 4, 가치 20), B(무게 6, 가치 18)인 분할 가능 배낭의 최대 가치를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 52 정답과 해설

용량 7, 물건 A(무게 4, 가치 20), B(무게 6, 가치 18)인 분할 가능 배낭의 최대 가치를 쓰시오.

29

해설

A를 전부 담고 B를 무게 3만큼 담는다. 20+18×3/6=29다.

문제 53

용량 5, 물건 A(2,8), B(4,16)인 분할 가능 배낭에서 A를 먼저 담을 때 B의 비율과 총가치는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 53 정답과 해설

용량 5, 물건 A(2,8), B(4,16)인 분할 가능 배낭에서 A를 먼저 담을 때 B의 비율과 총가치는?

B의 3/4, 총가치 20

해설

두 물건의 밀도가 모두 4라서 용량 5를 채우면 가치 20이다.

문제 54

무방향 그래프 간선 AB=1, BC=2, AC=3, CD=4, BD=5다. Kruskal의 선택 간선과 총비용은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 54 정답과 해설

무방향 그래프 간선 AB=1, BC=2, AC=3, CD=4, BD=5다. Kruskal의 선택 간선과 총비용은?

AB,BC,CD, 총 7

해설

AC는 A와 C가 이미 연결되어 사이클을 만들므로 거절한다.

문제 55

간선 AB=1, AC=2, BC=3인 삼각형을 Kruskal 순서로 모두 검사한다. 세 번째 간선을 거절하는 이유와 최종 비용은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 55 정답과 해설

간선 AB=1, AC=2, BC=3인 삼각형을 Kruskal 순서로 모두 검사한다. 세 번째 간선을 거절하는 이유와 최종 비용은?

B,C가 같은 연결 요소, 비용 3

해설

AB와 AC를 선택한 뒤 세 정점이 모두 연결되어 있다.

문제 56

정점 {A,B,C,D,E}, 간선 AB=1, BC=2, DE=3만 있다. 최소 신장 숲의 간선 수와 비용은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 56 정답과 해설

정점 {A,B,C,D,E}, 간선 AB=1, BC=2, DE=3만 있다. 최소 신장 숲의 간선 수와 비용은?

3개, 비용 6

해설

연결 요소는 {A,B,C}와 {D,E} 두 개다. V−2=3개 간선을 선택한다.

문제 57

간선 AB=−3, BC=1, AC=4인 연결 그래프의 MST 비용은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 57 정답과 해설

간선 AB=−3, BC=1, AC=4인 연결 그래프의 MST 비용은?

−2

해설

AB와 BC를 선택한다. 음수 가중치도 MST에 사용할 수 있다.

문제 58

분리 집합을 {0},{1},{2},{3}으로 시작한다. union(0,1), union(2,3), union(1,2) 뒤 집합 수와 find(0)==find(3)의 참거짓은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 58 정답과 해설

분리 집합을 {0},{1},{2},{3}으로 시작한다. union(0,1), union(2,3), union(1,2) 뒤 집합 수와 find(0)==find(3)의 참거짓은?

집합 1개, 참

해설

대표 번호 자체는 구현에 따라 달라도 모든 원소가 같은 집합이라는 결과는 같다.

문제 59

4정점 완전 그래프의 모든 간선 가중치가 2다. MST의 간선 수와 비용은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 59 정답과 해설

4정점 완전 그래프의 모든 간선 가중치가 2다. MST의 간선 수와 비용은?

3개, 비용 6

해설

어떤 신장 트리도 간선 3개를 가지므로 비용은 6이다. 선택 간선은 유일하지 않다.

문제 60

무방향 간선 AB=2, AC=3, BC=2다. A에서 C까지 MST 경로 길이와 실제 최단 거리는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 60 정답과 해설

무방향 간선 AB=2, AC=3, BC=2다. A에서 C까지 MST 경로 길이와 실제 최단 거리는?

MST 경로 4, 최단 거리 3

해설

MST는 AB,BC를 고른다. 최단 경로는 간선 AC 하나다.

문제 61

무방향 완전 그래프의 거리 AB=2, AC=9, AD=8, BC=3, BD=7, CD=4다. A에서 최근접 이웃 순회와 비용은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 61 정답과 해설

무방향 완전 그래프의 거리 AB=2, AC=9, AD=8, BC=3, BD=7, CD=4다. A에서 최근접 이웃 순회와 비용은?

A-B-C-D-A, 비용 17

해설

미방문 도시 중 가장 가까운 도시를 선택한 뒤 D에서 A로 돌아가는 8을 더한다.

문제 62

거리는 AB=2, AC=2, AD=9, BC=5, BD=3, CD=4다. A에서 출발하며 동률은 이름이 빠른 도시를 고른다. 최근접 이웃 순회는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 62 정답과 해설

거리는 AB=2, AC=2, AD=9, BC=5, BD=3, CD=4다. A에서 출발하며 동률은 이름이 빠른 도시를 고른다. 최근접 이웃 순회는?

A-B-D-C-A, 비용 11

해설

A에서는 B를 고르고 B에서는 D, D에서는 C를 고른다. 비용은 2+3+4+2다.

문제 63

방향 간선 S→A=4, S→B=1, B→A=2, A→C=1, B→C=5다. S에서 각 정점 A,B,C까지 최단 거리는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 63 정답과 해설

방향 간선 S→A=4, S→B=1, B→A=2, A→C=1, B→C=5다. S에서 각 정점 A,B,C까지 최단 거리는?

A=3, B=1, C=4

해설

B를 거쳐 A의 거리를 3으로 줄이고, A를 거쳐 C의 거리를 4로 줄인다.

문제 64

dist[u]=7, dist[v]=15이고 간선 u→v의 가중치가 3이다. 한 번 완화한 뒤 dist[v]는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 64 정답과 해설

dist[u]=7, dist[v]=15이고 간선 u→v의 가중치가 3이다. 한 번 완화한 뒤 dist[v]는?

10

해설

기존 15와 새 후보 7+3=10 중 작은 값을 저장한다.

문제 65

dist[u]=7, dist[v]=8이고 간선 u→v의 가중치가 3이다. 완화 후 dist[v]는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 65 정답과 해설

dist[u]=7, dist[v]=8이고 간선 u→v의 가중치가 3이다. 완화 후 dist[v]는?

8

해설

새 후보 10이 기존 8보다 크므로 갱신하지 않는다.

문제 66

최소 힙에서 (거리 9, 정점 A)를 꺼냈는데 dist[A]=5다. 이 항목을 이용한 이웃 완화 횟수는? 오래된 항목을 건너뛰는 구현이다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 66 정답과 해설

최소 힙에서 (거리 9, 정점 A)를 꺼냈는데 dist[A]=5다. 이 항목을 이용한 이웃 완화 횟수는? 오래된 항목을 건너뛰는 구현이다.

0회

해설

9>5이므로 이미 더 좋은 경로를 찾았다. 이 항목을 버리고 다음 항목을 꺼낸다.

문제 67

방향 간선 S→A=0, A→B=2만 있고 C는 고립되어 있다. S에서 [S,A,B,C]의 거리 배열은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 67 정답과 해설

방향 간선 S→A=0, A→B=2만 있고 C는 고립되어 있다. S에서 [S,A,B,C]의 거리 배열은?

[0,0,2,∞]

해설

가중치 0인 간선도 유효하다. C에는 도달할 수 없다.

문제 68

X 중간고사 출제 제외

모든 작업은 시각 0에 도착한다. 처리 시간 [3,1,4,2]에서 완료 시각 합을 최소화하는 처리 시간 순서와 합을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 68 정답과 해설

X 중간고사 출제 제외

모든 작업은 시각 0에 도착한다. 처리 시간 [3,1,4,2]에서 완료 시각 합을 최소화하는 처리 시간 순서와 합을 쓰시오.

순서 [1,2,3,4], 합 20

해설

누적 완료 시각은 1,3,6,10이며 합은 20이다.

문제 69

X 중간고사 출제 제외

처리 시간 [2,5,1]을 입력 순서대로 수행한 완료 시각 합과 최솟값을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 69 정답과 해설

X 중간고사 출제 제외

처리 시간 [2,5,1]을 입력 순서대로 수행한 완료 시각 합과 최솟값을 쓰시오.

입력 순서 17, 최솟값 12

해설

입력 순서는 2+7+8=17이다. 정렬 순서는 1+3+8=12다.

문제 70

X 중간고사 출제 제외

작업 시간 [4,4,4]를 어떤 순서로 처리해도 완료 시각 합은 얼마인가?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 70 정답과 해설

X 중간고사 출제 제외

작업 시간 [4,4,4]를 어떤 순서로 처리해도 완료 시각 합은 얼마인가?

24

해설

완료 시각은 4,8,12이므로 합은 24다. 같은 처리 시간끼리의 순서는 영향을 주지 않는다.

문제 71

동전 {1,3,4}, 금액 6을 이용해 큰 동전 우선이 일반 화폐에서 항상 최적이라는 주장을 반박하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 71 정답과 해설

동전 {1,3,4}, 금액 6을 이용해 큰 동전 우선이 일반 화폐에서 항상 최적이라는 주장을 반박하시오.

탐욕은 4+1+1로 3개, 최적은 3+3으로 2개다.

해설

모든 동전보다 금액 6이 크므로 한 개로는 만들 수 없다. 두 개짜리 해를 제시하면 최적성까지 확인된다.

문제 72

동전 {1,5,10,50}에서 큰 동전 우선의 최소 개수를 교환 관점으로 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 72 정답과 해설

동전 {1,5,10,50}에서 큰 동전 우선의 최소 개수를 교환 관점으로 설명하시오.

1원 5개는 5원으로, 5원 2개는 10원으로, 10원 5개는 50원으로 바꿔 개수를 줄일 수 있다.

해설

최적해에는 교환 가능한 작은 동전 묶음이 남지 않는다. 이 제한이 큰 동전부터 몫을 취한 표현과 일치한다.

문제 73

활동 수 최대화에서 시작 시각이 빠른 순서가 실패하는 양의 길이 구간 세 개를 만들고 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 73 정답과 해설

활동 수 최대화에서 시작 시각이 빠른 순서가 실패하는 양의 길이 구간 세 개를 만들고 설명하시오.

A[0,10), B[1,2), C[2,3). A를 먼저 고르면 1개, B,C는 2개다.

해설

일찍 시작하는 긴 활동이 이후의 여러 활동을 막을 수 있다. 끝점이 닿는 것은 허용한다.

문제 74

활동 선택에서 종료 시각이 가장 빠른 활동을 최적해의 첫 활동과 바꿔도 되는 이유를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 74 정답과 해설

활동 선택에서 종료 시각이 가장 빠른 활동을 최적해의 첫 활동과 바꿔도 되는 이유를 쓰시오.

탐욕 활동은 최적해의 첫 활동보다 늦게 끝나지 않으므로 뒤의 활동들과 충돌하지 않는다.

해설

첫 활동 하나를 바꿔도 개수는 같고 남은 일정이 유지된다. 따라서 탐욕 활동을 포함하는 최적해가 존재한다.

문제 75

활동 선택 코드가 거절한 활동의 종료 시각으로 last_end를 갱신한다. [0,2), [1,10), [2,3)을 이용해 오류를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 75 정답과 해설

활동 선택 코드가 거절한 활동의 종료 시각으로 last_end를 갱신한다. [0,2), [1,10), [2,3)을 이용해 오류를 설명하시오.

첫 활동 선택 후 두 번째를 거절해도 last_end는 2여야 한다. 10으로 바꾸면 세 번째를 잘못 거절한다.

해설

last_end는 마지막 검사 활동이 아니라 마지막 선택 활동을 나타내야 한다.

문제 76

가치 밀도 탐욕이 0/1 배낭에서 실패하는 예를 제시하고 분할 가능 문제와의 차이를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 76 정답과 해설

가치 밀도 탐욕이 0/1 배낭에서 실패하는 예를 제시하고 분할 가능 문제와의 차이를 설명하시오.

용량 50, (10,60),(20,100),(30,120). 탐욕 160, 최적 220이다.

해설

분할 가능하면 마지막 물건 일부를 담아 240을 얻지만 0/1에서는 남은 공간에 일부만 넣을 수 없다.

문제 77

분할 가능 배낭에서 낮은 밀도 물건을 먼저 담은 해를 개선하는 교환을 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 77 정답과 해설

분할 가능 배낭에서 낮은 밀도 물건을 먼저 담은 해를 개선하는 교환을 설명하시오.

담긴 낮은 밀도 무게 ε를 빼고 아직 남은 높은 밀도 물건 무게 ε를 넣는다.

해설

총무게는 같고 가치는 ε×밀도 차이만큼 증가한다. 높은 밀도 물건이 남았는데 낮은 밀도를 먼저 쓰는 해는 최적일 수 없다.

문제 78

분할 가능 배낭 구현에서 남은 용량 r보다 물건 무게 w가 큰데 가치 v를 전부 더했다. 올바른 계산을 쓰고 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 78 정답과 해설

분할 가능 배낭 구현에서 남은 용량 r보다 물건 무게 w가 큰데 가치 v를 전부 더했다. 올바른 계산을 쓰고 이유를 설명하시오.

추가 가치는 v×r/w이고 추가 무게는 r이다.

해설

물건의 r/w 비율만 담았으므로 가치도 같은 비율만 인정해야 한다.

문제 79

Kruskal이 같은 연결 요소의 두 정점을 잇는 간선을 거절하는 이유를 경로를 이용해 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 79 정답과 해설

Kruskal이 같은 연결 요소의 두 정점을 잇는 간선을 거절하는 이유를 경로를 이용해 설명하시오.

두 정점 사이에는 이미 선택 간선만으로 경로가 있다. 새 간선을 더하면 그 경로와 함께 사이클이 생긴다.

해설

신장 트리는 사이클이 없어야 한다. 서로 다른 요소를 연결하는 경우에는 이런 기존 경로가 없다.

문제 80

정점 A,B,C와 간선 AB(1), BC(2), AC(3)이 있는 무방향 그래프에 Kruskal을 적용한다. 가중치가 작은 순서로 간선을 확인할 때 선택하는 간선과 총비용을 쓰고, 마지막 간선을 고르지 않는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 80 정답과 해설

정점 A,B,C와 간선 AB(1), BC(2), AC(3)이 있는 무방향 그래프에 Kruskal을 적용한다. 가중치가 작은 순서로 간선을 확인할 때 선택하는 간선과 총비용을 쓰고, 마지막 간선을 고르지 않는 이유를 설명하시오.

AB와 BC를 선택하고 총비용은 3이다. AC를 더하면 사이클이 생긴다.

해설

AB로 A,B를 연결하고 BC로 C까지 연결한다. 이미 연결된 A,C 사이의 간선을 추가할 필요가 없으며 추가하면 A-B-C-A 사이클이 생긴다.

문제 81

같은 가중치의 간선이 있으면 MST가 반드시 여러 개라는 주장이 틀림을 보이시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 81 정답과 해설

같은 가중치의 간선이 있으면 MST가 반드시 여러 개라는 주장이 틀림을 보이시오.

간선 AB=1, BC=1만 있는 3정점 경로는 신장 트리가 자기 자신 하나뿐이다.

해설

가중치가 모두 다르면 유일하다는 충분조건의 역은 성립하지 않는다.

문제 82

정점 V개, 연결 요소 c개인 무방향 그래프의 최소 신장 숲 간선 수를 유도하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 82 정답과 해설

정점 V개, 연결 요소 c개인 무방향 그래프의 최소 신장 숲 간선 수를 유도하시오.

V−c개다.

해설

각 연결 요소의 정점 수를 v_i라 하면 트리 간선은 v_i-1개다. 합하면 $\Sigma v_i - c = V - c$다.

문제 83

최근접 이웃 TSP에서 출발점으로 돌아오는 간선을 빠뜨리면 어떤 문제를 계산하는지 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 83 정답과 해설

최근접 이웃 TSP에서 출발점으로 돌아오는 간선을 빠뜨리면 어떤 문제를 계산하는지 설명하시오.

모든 도시를 한 번씩 방문하는 경로의 비용만 계산하게 된다.

해설

TSP 순회는 마지막 도시와 출발 도시도 연결해야 한다. 누락한 귀환 비용만큼 순회 비용을 과소평가한다.

문제 84

최근접 이웃 TSP의 결과가 재현되도록 구현 명세에 추가할 두 조건과 그 이유를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 84 정답과 해설

최근접 이웃 TSP의 결과가 재현되도록 구현 명세에 추가할 두 조건과 그 이유를 쓰시오.

출발 도시와 거리가 같은 후보 사이의 선택 규칙을 명시한다.

해설

두 조건이 바뀌면 방문 순서와 총비용이 달라질 수 있다. 예를 들어 동률이면 도시 번호가 작은 쪽을 선택하도록 정한다.

문제 85

방향 간선 S→A=2, S→B=5, B→A=−4다. 이미 확정한 정점의 거리를 다시 갱신하지 않는 Dijkstra가 S에서 A까지의 최단 거리를 틀리는 이유를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 85 정답과 해설

방향 간선 S→A=2, S→B=5, B→A=−4다. 이미 확정한 정점의 거리를 다시 갱신하지 않는 Dijkstra가 S에서 A까지의 최단 거리를 틀리는 이유를 쓰시오.

A를 거리 2로 먼저 확정하지만 실제 최단 거리는 S-B-A의 1이다.

해설

음수 간선으로 인해 나중에 확정한 B를 거치는 경로가 A의 거리를 더 줄인다. 이 그래프에는 사이클이 없어도 오류가 생긴다.

문제 86

Dijkstra에서 미확정 정점 중 최소 잠정 거리 정점을 확정할 수 있는 이유를 비음수 조건과 연결해 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 86 정답과 해설

Dijkstra에서 미확정 정점 중 최소 잠정 거리 정점을 확정할 수 있는 이유를 비음수 조건과 연결해 설명하시오.

그보다 긴 잠정 거리의 미확정 정점을 거쳐 추가 비음수 간선을 지나도 더 짧은 경로를 만들 수 없다.

해설

이미 확정한 정점의 거리가 정확하다는 불변식과 경로가 확정 영역을 처음 벗어나는 간선을 함께 사용한다.

문제 87

인접 행렬에서 0을 간선 없음으로 표현하면 가중치 0인 간선을 처리할 수 없다. 수정 방법과 주의점을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 87 정답과 해설

인접 행렬에서 0을 간선 없음으로 표현하면 가중치 0인 간선을 처리할 수 없다. 수정 방법과 주의점을 쓰시오.

간선 없음은 −1 또는 별도의 존재 배열로 표시하고 0은 유효한 가중치로 남긴다.

해설

Dijkstra 입력은 비음수 가중치이므로 −1을 부재 표식으로 구분할 수 있다. 거리의 ∞와 간선 부재 표식도 구분한다.

문제 88

최단 거리의 ∞를 큰 정수로 표현할 때 완화 전에 확인할 조건 두 가지를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 88 정답과 해설

최단 거리의 ∞를 큰 정수로 표현할 때 완화 전에 확인할 조건 두 가지를 쓰시오.

출발 정점까지 도달 가능한지 확인하고, dist[u]+w가 자료형 범위를 넘지 않도록 한다.

해설

도달 불가 표식에 가중치를 더하거나 정수 오버플로가 나면 가짜로 작은 거리가 생길 수 있다. 입력 상한에 맞는 INF와 자료형을 선택한다.

문제 89

X 중간고사 출제 제외

모든 작업이 시각 0에 도착한다. 긴 작업 a가 짧은 작업 b 앞에 있을 때 인접 교환으로 완료 시각 합이 줄어드는 이유를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 89 정답과 해설

X 중간고사 출제 제외

모든 작업이 시각 0에 도착한다. 긴 작업 a가 짧은 작업 b 앞에 있을 때 인접 교환으로 완료 시각 합이 줄어드는 이유를 쓰시오.

a>b일 때 두 작업의 합은 앞 순서 2a+b, 교환 후 2b+a여서 a−b만큼 줄어든다.

해설

앞부분의 공통 시작 시각은 상쇄되고 두 작업 이후의 시작 시각은 총길이 a+b가 같아 변하지 않는다.

문제 90

예제 100개에서 탐욕 결과가 완전 탐색과 같았다. 이것만으로 모든 입력의 최적성이 증명되는지 설명하고 더 필요한 작업을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 90 정답과 해설

예제 100개에서 탐욕 결과가 완전 탐색과 같았다. 이것만으로 모든 입력의 최적성이 증명되는지 설명하고 더 필요한 작업을 쓰시오.

증명되지 않는다. 입력 전체를 포괄하는 교환 논증 등 정당성 증명이 필요하다.

해설

작은 입력의 완전 탐색 비교는 반례를 찾는 검증 도구로 유용하지만 시험한 범위 밖까지 보장하지는 않는다.

문제 91

coin_count(amount)는 {500,100,50,10,1} 동전을 무제한 사용해 amount를 만드는 최소 동전 수를 반환한다. $0 \le amount \le 10^9$. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 91 정답과 해설

정수배 화폐 거스름돈

내림차순으로 몫을 개수에 더하고 나머지를 다음 종류로 넘긴다.

해설

고정된 5종류이므로 시간 O(1). 일반 m종류에서는 시간 O(m).

문제 91 Python 구현

정수배 화폐 거스름돈

```python
def coin_count(amount):
    count = 0
    for coin in [500, 100, 50, 10, 1]:
        count += amount // coin
        amount %= coin
    return count
```

고정된 5종류이므로 시간 O(1). 일반 m종류에서는 시간 O(m).

문제 91 C 구현

정수배 화폐 거스름돈

```c
int coin_count(int amount) {
    int coins[]={500,100,50,10,1}, count=0;
    for (int i=0; i<5; ++i) {
        count+=amount/coins[i];
        amount%=coins[i];
    }
    return count;
}
```

고정된 5종류이므로 시간 O(1). 일반 m종류에서는 시간 O(m).

문제 92

activities는 (시작,종료) 목록이며 종료 시각 오름차순으로 정렬되어 있다. $0 \le$ 시작 $<$ 종료 $\le 10^9$, $0 \le n \le 1000$. [시작,종료) 기준 최대 활동 수 activity_count를 반환한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 92 정답과 해설

활동 선택

마지막 선택의 종료 시각 이상에서 시작하는 활동만 선택한다.

해설

정렬된 입력에서 시간 O(n). 정렬까지 하면 O(n log n).

문제 92 Python 구현

활동 선택

```python
def activity_count(activities):
    last = 0
    count = 0
    for start, end in activities:
        if start >= last:
            count += 1
            last = end
    return count
```

정렬된 입력에서 시간 O(n). 정렬까지 하면 O(n log n).

문제 92 C 구현

활동 선택

```c
int activity_count(const int start[], const int end[], int n) {
    int last=0, count=0;
    for (int i=0; i<n; ++i) {
        if (start[i]>=last) {
            ++count;
            last=end[i];
        }
    }
    return count;
}
```

정렬된 입력에서 시간 O(n). 정렬까지 하면 O(n log n).

문제 93

(무게,가치) 물건 목록은 가치/무게 내림차순이다. 무게·가치는 $1..10^6$, $n \le 1000$, 용량 $0..10^6$. fractional_value가 최대 가치를 실수로 반환하게 하시오. 오차 10^{-6} 허용. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 93 정답과 해설

분할 가능 배낭

남은 용량과 물건 무게의 최솟값만큼 담고 비례 가치를 더한다.

해설

정렬된 입력에서 시간 O(n). 나누기 전에 실수 연산으로 전환한다.

문제 93 Python 구현

분할 가능 배낭

```python
def fractional_value(items, capacity):
    value = 0.0
    for weight, price in items:
        take = min(capacity, weight)
        value += price * (take / weight)
        capacity -= take
        if capacity == 0:
            break
    return value
```

정렬된 입력에서 시간 O(n). 나누기 전에 실수 연산으로 전환한다.

문제 93 C 구현

분할 가능 배낭

```c
double fractional_value(const int w[], const int v[],
                        int n, int capacity) {
    double value=0.0;
    for (int i=0; i<n && capacity>0; ++i) {
        int take=capacity<w[i] ? capacity:w[i];
        value+=(double)v[i]*take/w[i];
        capacity-=take;
    }
    return value;
}
```

정렬된 입력에서 시간 O(n). 나누기 전에 실수 연산으로 전환한다.

문제 94

X 코딩 실습 제외

parent는 0..n−1 정점의 유효한 분리 집합 숲이다. 루트는 parent[r]=r이다. n≤1000. Python의 재귀 제한은 충분히 크게 설정되어 있다. dsu_find(parent,x)는 대표를 반환하고 방문 경로를 압축한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 94 정답과 해설

X 코딩 실습 제외

경로 압축 find

루트까지 재귀로 이동한 뒤 돌아오며 각 부모를 대표로 바꾼다.

해설

한 호출의 최악 시간 O(n). 경로 압축만으로 모든 단일 호출을 O(1)이라 할 수 없다. Python 재귀 제한은 입력 깊이 이상으로 설정한다.

문제 94 Python 구현

X 코딩 실습 제외

경로 압축 find

```python
def dsu_find(parent, x):
    if parent[x] != x:
        parent[x] = dsu_find(parent, parent[x])
    return parent[x]
```

한 호출의 최악 시간 O(n). 경로 압축만으로 모든 단일 호출을 O(1)이라 할 수 없다. Python 재귀 제한은 입력 깊이 이상으로 설정한다.

문제 94 C 구현

X 코딩 실습 제외

경로 압축 find

```c
int dsu_find(int parent[], int x) {
    if (parent[x]!=x)
        parent[x]=dsu_find(parent,parent[x]);
    return parent[x];
}
```

한 호출의 최악 시간 O(n). 경로 압축만으로 모든 단일 호출을 O(1)이라 할 수 없다. Python 재귀 제한은 입력 깊이 이상으로 설정한다.

문제 95

1≤n≤1000인 무방향 그래프의 간선 (u,v,w)은 가중치 오름차순이다. $-10^6 \le w \le 10^6$. kruskal_cost는 MST 비용을 반환한다. 연결 불가면 -10^{18}이다. dsu_find는 제공된다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 95 정답과 해설

Kruskal 최소 신장 비용

서로 다른 집합을 잇는 간선만 선택한다. 선택 수 n−1을 확인한다.

해설

1≤n≤1000. 이 구현은 랭크 병합이 없다. 보수적 시간 상한 O(n+mn).

문제 95 Python 구현

Kruskal 최소 신장 비용

```python
def kruskal_cost(n, edges):
    parent = list(range(n))
    cost = used = 0
    for u, v, w in edges:
        ru, rv = dsu_find(parent,u), dsu_find(parent,v)
        if ru != rv:
            parent[ru] = rv
            cost += w
            used += 1
    return cost if used == n - 1 else -10**18
```

1≤n≤1000. 이 구현은 랭크 병합이 없다. 보수적 시간 상한 O(n+mn).

문제 95 C 구현

Kruskal 최소 신장 비용

```c
long long kruskal_cost(int n, const int edges[][3], int m) {
    int parent[1000], used=0;
    long long cost=0;
    for (int i=0; i<n; ++i) parent[i]=i;
    for (int i=0; i<m; ++i) {
        int ru=dsu_find(parent,edges[i][0]);
        int rv=dsu_find(parent,edges[i][1]);
        if (ru!=rv) {
            parent[ru]=rv; cost+=edges[i][2]; ++used;
        }
    }
    return used==n-1 ? cost:-1000000000000000000LL;
}
```

1≤n≤1000. 이 구현은 랭크 병합이 없다. 보수적 시간 상한 O(n+mn).

문제 96

X 코딩 실습 제외

거리 dist와 확정 여부 used가 주어진다. 미확정이며 거리가 INF 미만인 정점 중 (거리,번호)가 가장 작은 번호를 pick_vertex로 반환한다. 없으면 −1. n≤100, INF=10^{18}. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 96 정답과 해설

X 코딩 실습 제외

Dijkstra 최소 정점 선택

번호 오름차순으로 순회하며 더 작은 거리에서만 후보를 교체한다.

해설

시간 O(n). 동률에서 기존 번호를 유지한다.

문제 96 Python 구현

X 코딩 실습 제외

Dijkstra 최소 정점 선택

```python
def pick_vertex(dist, used):
    best = -1
    for i in range(len(dist)):
        if not used[i] and dist[i] < 10**18:
            if best == -1 or dist[i] < dist[best]:
                best = i
    return best
```

시간 O(n). 동률에서 기존 번호를 유지한다.

문제 96 C 구현

X 코딩 실습 제외

Dijkstra 최소 정점 선택

```
int pick_vertex(const long long dist[], const int used[], int n) {
    int best=-1;
    for (int i=0; i<n; ++i) {
        if (!used[i] && dist[i]<1000000000000000000LL) {
            if (best==-1 || dist[i]<dist[best]) best=i;
        }
    }
    return best;
}
```

시간 O(n). 동률에서 기존 번호를 유지한다.

문제 97

X 코딩 실습 제외

1≤n≤100, 0≤s<n. g[u][v]는 방향 간선 가중치 0..10^6이며 −1은 간선 없음이다. dijkstra(g,s)는 거리 배열을 구한다. 도달 불가는 10^{18}. C 열 폭은 100. pick_vertex는 제공된다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 97 정답과 해설

X 코딩 실습 제외

인접 행렬 Dijkstra

가장 짧은 미확정 정점을 선택하고 존재하는 간선을 완화한다.

해설

시간 $O(n^2)$. 주어진 상한에서 거리 덧셈은 long long에 안전하다.

문제 97 Python 구현

X 코딩 실습 제외

인접 행렬 Dijkstra

```python
def dijkstra(g, s):
    n = len(g)
    dist, used = [10**18]*n, [False]*n
    dist[s] = 0
    for _ in range(n):
        u = pick_vertex(dist, used)
        if u == -1:
            break
        used[u] = True
        for v in range(n):
            if g[u][v] >= 0:
                dist[v] = min(dist[v], dist[u] + g[u][v])
    return dist
```

시간 $O(n^2)$. 주어진 상한에서 거리 덧셈은 long long에 안전하다.

문제 97 C 구현

X 코딩 실습 제외

인접 행렬 Dijkstra

```
01  void dijkstra(const int g[][100], int n, int s, long long dist[]) {
02      int used[100]={0};
03      for (int i=0; i<n; ++i) dist[i]=1000000000000000000LL;
04      dist[s]=0;
05      for (int step=0; step<n; ++step) {
06          int u=pick_vertex(dist,used,n);
07          if (u==-1) break;
08          used[u]=1;
09          for (int v=0; v<n; ++v) {
10              if (g[u][v]>=0 && dist[u]+g[u][v]<dist[v])
11                  dist[v]=dist[u]+g[u][v];
12          }
13      }
14  }
```

시간 $O(n^2)$. 주어진 상한에서 거리 덧셈은 long long에 안전하다.

문제 98

X 코딩 실습 제외

n=2..50인 완전 무방향 거리 행렬 d, 가중치 1..10^6(대각선 0)이다. 0에서 시작, 동률은 작은 도시 번호로 고른다. nearest_tour는 귀환을 포함한 휴리스틱 비용을 반환한다. C 열 폭은 50. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 98 정답과 해설

X 코딩 실습 제외

최근접 이웃 TSP

매번 가장 가까운 미방문 도시로 이동하고 마지막에 0으로 돌아온다.

해설

시간 $O(n^2)$. 최적 TSP 비용을 보장하지 않는다.

문제 98 Python 구현

X 코딩 실습 제외

최근접 이웃 TSP

```python
def nearest_tour(d):
    n = len(d)
    used = [False]*n
    used[0] = True
    u = cost = 0
    for _ in range(n - 1):
        v = min((i for i in range(n) if not used[i]),
                key=lambda i: (d[u][i], i))
        cost += d[u][v]
        used[v] = True
        u = v
    return cost + d[u][0]
```

시간 $O(n^2)$. 최적 TSP 비용을 보장하지 않는다.

문제 98 C 구현

X 코딩 실습 제외

최근접 이웃 TSP

```
01  long long nearest_tour(const int d[][50], int n) {
02      int used[50]={0}, u=0;
03      long long cost=0; used[0]=1;
04      for (int step=1; step<n; ++step) {
05          int v=-1;
06          for (int i=0; i<n; ++i)
07              if (!used[i] && (v==-1 || d[u][i]<d[u][v])) v=i;
08          cost+=d[u][v]; used[v]=1; u=v;
09      }
10      return cost+d[u][0];
11  }
```

시간 $O(n^2)$. 최적 TSP 비용을 보장하지 않는다.

문제 99

X 중간고사 출제 제외

모두 시각 0에 도착한 작업의 처리 시간 a가 오름차순이다. $0 \le n \le 1000$, 시간 $1..10^6$. completion_sum은 이 순서의 완료 시각 합을 반환한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 99 정답과 해설

X 중간고사 출제 제외

완료 시각 합

처리 시간 누적합을 구하면서 각 누적합을 다시 더한다.

해설

정렬된 입력에서 시간 O(n). 최대 합 500500000000은 int 범위를 넘는다.

문제 99 Python 구현

X 중간고사 출제 제외

완료 시각 합

```python
def completion_sum(a):
    elapsed = total = 0
    for duration in a:
        elapsed += duration
        total += elapsed
    return total
```

정렬된 입력에서 시간 O(n). 최대 합 500500000000은 int 범위를 넘는다.

문제 99 C 구현

X 중간고사 출제 제외

완료 시각 합

```c
long long completion_sum(const int a[], int n) {
    long long elapsed=0, total=0;
    for (int i=0; i<n; ++i) {
        elapsed+=a[i];
        total+=elapsed;
    }
    return total;
}
```

정렬된 입력에서 시간 O(n). 최대 합 500500000000은 int 범위를 넘는다.

문제 100

relax(dist,u,v,w)는 u에서 v로 가는 가중치 w 간선을 완화하고 갱신했으면 1, 아니면 0을 반환한다. INF=10^{18}, 유한 거리≤10^{12}, 0≤w≤10^{6}이다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 100 정답과 해설

간선 완화와 변경 여부

u가 도달 가능하고 새 후보가 더 작을 때만 거리와 변경 여부를 갱신한다.

해설

시간 O(1). 같은 거리 후보는 갱신으로 세지 않는다.

문제 100 Python 구현

간선 완화와 변경 여부

```python
def relax(dist, u, v, w):
    if dist[u] == 10**18:
        return 0
    candidate = dist[u] + w
    if candidate < dist[v]:
        dist[v] = candidate
        return 1
    return 0
```

시간 O(1). 같은 거리 후보는 갱신으로 세지 않는다.

문제 100 C 구현

간선 완화와 변경 여부

```c
int relax(long long dist[], int u, int v, int w) {
    if (dist[u]==1000000000000000000LL) return 0;
    long long candidate=dist[u]+w;
    if (candidate<dist[v]) {
        dist[v]=candidate;
        return 1;
    }
    return 0;
}
```

시간 O(1). 같은 거리 후보는 갱신으로 세지 않는다.