연습문제

1 리스트 A, L, G, O, R, I, T, H, M을 선택 정렬을 이용해 오름차순으로 정렬하라.

2 다음 리스트를 선택 정렬을 이용해 오름차순으로 정렬하라. 각 단계에서의 배열의 내용을 나타내어라.

7	4	9	6	3	8	7	5

3 선택 정렬이 안정성을 만족하지 않는 경우의 예를 보여라.

4* 선택 정렬의 기본 전략을 그대로 이용하면서 안정성을 만족하도록 알고리즘을 수정하는 방법을 설명하라.

5 n개의 검은 구슬과 n개의 흰 구슬이 무작위로 일렬로 나열되어 있다. 이 구슬들을 흰 구슬이 모두 먼저 나오고 이후에 검은 구슬이 나오도록 정렬하려고 한다. 이때, 구슬의 이동은 인접한 두 구슬을 교환하는 것만 가능하다. 이 문제를 해결하기 위한 알고리즘을 작성하라.

6 만약 리스트가 정렬되어 있다면 순차 탐색을 어떻게 개선할 수 있을까? 알고리즘 3.2를 수정해 보라.

7* 알고리즘 3.2의 순차 탐색은 매 반복에서 두 가지를 비교해야 한다. if A[i] == key는 코드에 보이는 비교 연산이고, 보이지는 않지만 반드시 필요한 비교도 있다. for 문에서 인덱스 i가 len(A)보다 작은지를 매번 검사하는 것인데, 이때 비교 연산이 사용된다. 그런데 순차 탐색의 알고리즘을 약간 바꾸면 이 비교 연산을 피할 수 있다. 리스트의 맨 마지막에 찾고자 하는 key값을 추가로 저장해 두는 것이다(이것을 "보초병(sentinel)"을 세운다고 말한다). 이렇게 되면 항상 리스트에는 찾는 값이 있으므로 리스트의 모든 항목을 검사했는지를 매 반복에서 검사할 필요가 없다. for 문 대신에 while을 사용하여 이 알고리즘을 구현해 보라.

8 100개의 0으로 구성된 이진 텍스트(binary text)에서 다음의 패턴을 찾으려고 한다. 알고리즘 3.3과 같은 억지 기법을 사용한다면 각각 몇 번의 비교 연산이 필요한가?

(1) 00001 (2) 10000 (3) 01010

9 길이가 n인 텍스트에서 길이가 m인 패턴을 알고리즘 3.3을 이용해 찾으려고 한다. 최악의 입력을 설명하라. 정확히 몇 번의 비교 연산이 필요한가?

10 문자열 매칭 알고리즘에서 만약 패턴의 왼쪽 문자부터 오른쪽이 아니라 맨 오른쪽 문자부터 왼쪽 방향으로 검사한다면 유리한 점이 있을까?

11* 입력으로 주어진 문자열에서 A로 시작하고 B로 끝나는 부분 문자열(substring)의 개수를 구하는 문제가 주어졌다. 예를 들어, 문자열 ADBAAEDBA에는 4개의 부분 문자열이 있다. ADBAAEDBA, ADBAAEDBA, ADBAAEDBA, ADBAAEDBA이다.

(1) 이 문제에 대한 억지 기법 알고리즘을 설계하고, 시간 복잡도를 계산하라.

(2) 이 문제에 대한 더 효율적인 알고리즘을 찾아보라.

12* 알고리즘 3.4를 일차원 공간의 점에 적용하고자 한다. 즉, 각 점은 x_i와 같이 x 좌표로만 표시된다. n개의 점이 주어졌을 때 물음에 답하라.

(1) distance() 함수는 어떻게 수정되어야 할까?

(2) 알고리즘 3.4보다 더 효율적인 알고리즘을 설계해 보라.

13 유클리드 거리 외에도 평면상의 두 점의 거리에 대한 다양한 정의가 있다. 이러한 거리를 찾아보고, 유클리드 거리와 비교하라.

14 알고리즘 3.4는 k차원으로 확장할 수 있다. k차원의 두 점 $X=(x_1, x_2, \cdots, x_k)$와 $Y=(y_1, y_2, \cdots, y_k)$의 유클리드 거리는 다음과 같이 정의된다.

$$d(X, Y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

k차원의 최근접쌍 거리 알고리즘의 시간 복잡도를 설명하라.

15* 그래프 $G=(V, E)$가 주어졌을 때, 하나의 해밀토니안 사이클을 찾는 알고리즘을 억지 기법으로 설계하라. 간단한 그래프에 대해 이 알고리즘을 테스트하고 결과를 출력하라.

16* 15번 문제에서 모든 해밀토니안 사이클을 구하도록 알고리즘을 수정하라. 간단한 그래프에 대해 이 알고리즘을 테스트하고 결과를 출력하라.

17* 16번 문제에서 설계한 알고리즘을 바탕으로 외판원 문제에 대한 억지 기법 알고리즘을 작성하고 간단한 그래프에 대해 테스트하라.

18* 배낭 채우기 문제에 대한 억지 기법 알고리즘을 파이썬으로 구현하고 간단한 입력에 대해 테스트하라.

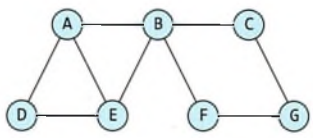
19* 일 배정 문제에 대한 억지 기법 알고리즘을 파이썬으로 구현하고 간단한 입력에 대해 테스트하라.

20* 일 배정 문제에 대한 헝가리안 알고리즘(Hungarian algorithm)을 찾아보라. 알고리즘을 구현하고, 시간 복잡도를 분석하라.

21 정렬 문제에 대해 완전 탐색을 적용할 수 있는 방법을 제시해 보라. 이 방법의 시간 복잡도를 분석하라.

22* 분할(partition) 문제는 주어진 집합 S를 분할하여 두 개의 부분집합으로 나누었을 때 원소들의 합이 같은 분할이 가능한지를 판단하는 문제이다. 이 문제에 대한 억지 기법 알고리즘을 작성해 보고, 시간 복잡도를 분석하라.

23* 다음과 같은 그래프가 주어졌다. 물음에 답하라.



(1) 이 그래프를 인접 행렬로 표현하라.

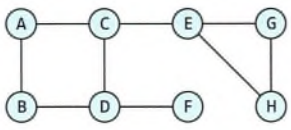
(2) 이 그래프를 인접 리스트로 표현하라.

(3) 정점 A를 출발 정점으로 하고 깊이 우선 방식으로 탐색하였을 때 방문하는 정점을 순서대로 나열하라. 단, 인접한 정점들 중에서는 알파벳으로 먼저 나오는 정점을 먼저 탐색한다고 가정하라.

(4) 위 문제에서 더 이상 방문하지 않은 인접 정점이 없어 첫 번째로 되돌아가는 정점을 찾아라.

(5) 정점 A를 출발 정점으로 하고 너비 우선 방식으로 탐색하였을 때 방문하는 정점을 순서대로 나열하라. 단, 인접한 정점들 중에서는 알파벳으로 먼저 나오는 정점을 큐에 먼저 넣는다고 가정하라.

24 그래프가 다음과 같이 인접 행렬로 표현되어 있다. 물음에 답하라.




```
vertex = [ 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H' ]
adjMat = [[ 0, 1, 1, 0, 0, 0, 0, 0 ],
          [ 1, 0, 0, 1, 0, 0, 0, 0 ],
          [ 1, 0, 0, 1, 1, 0, 0, 0 ],
          [ 0, 1, 1, 0, 0, 1, 0, 0 ],
          [ 0, 0, 1, 0, 0, 0, 1, 1 ],
          [ 0, 0, 0, 1, 0, 0, 0, 0 ],
          [ 0, 0, 0, 0, 1, 0, 0, 1 ],
          [ 0, 0, 0, 0, 1, 0, 1, 0 ]]
```

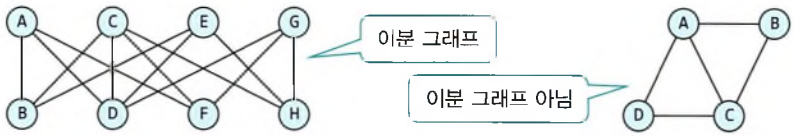
(1) 이 그래프에 대한 깊이 우선 탐색 알고리즘을 구현하라.

(2) 이 그래프에 대한 너비 우선 탐색 알고리즘을 구현하라.

25 어떤 희소 그래프의 정점의 수를 n이라 하고 간선의 수를 e라 하자. 만약 $e \in O(n)$이라면 인접 행렬을 사용한 DFS 알고리즘과 인접 리스트를 사용한 DFS 알고리즘 중에서 어느 방법이 시간적으로 더 효율적인가?

26* 신장 트리(spanning tree)란 그래프 내의 모든 정점을 포함하는 트리다. 모든 정점들이 연결되어 있고 사이클이 없어야 하며, 그래프의 n개의 정점을 정확히 $(n-1)$개의 간선으로 연결해야 한다. 깊이 우선 탐색이나 너비 우선 탐색을 이용하면 신장 트리를 구할 수 있다. 인접 행렬로 표현된 그래프에 대해 신장 트리를 구하는 알고리즘을 설계하라. 단, 깊이 우선 탐색을 이용하라.

27* 이분 그래프(bipartite graph)는 정점의 집합을 두 개의 부분집합 X와 Y로 분할할 수 있고, 모든 간선의 한 끝점은 X에 다른 끝점은 Y에 속하도록 할 수 있는 그래프를 말한다. 예를 들어, 다음은 이분 그래프와 그렇지 않은 그래프의 예를 보여준다. 물음에 답하라.



(1) 깊이 우선 탐색을 이용해 주어진 그래프가 이분 그래프인지를 판단하는 알고리즘을 설계하라.

(2) 너비 우선 탐색을 이용해 주어진 그래프가 이분 그래프인지를 판단하는 알고리즘을 설계하라.