**주의사항**

* 코드 구동 환경은 C++20 표준입니다.
* 답안 작성 시 정점, 간선 표현 방법과 높이, 깊이 등 정의는 수업 시간 표기를 기준으로 합니다.

**PART Ⅰ.**

1. C++ 컨테이너 queue를 이용해 자료구조 객체 Q를 만들고, 정수형 자료 1, 2, 3, 4, 5를 넣었다. 아래 명령을 실행한 후 Q의 요소(element)를 차례로 쓰라(1점).

|  |
| --- |
| Q.front(); |

1. 아래 코드(헤더 등 생략)에서 함수 funA를 호출하여 배열 arr의 요소를 오름차순으로 정렬하고, 콘솔창 출력값을 35689로 변경하려고 한다. 필요 시 ①을 수정하여 답에 ① 라인을 쓰고, 필요 없다면 “필요 없다”고 쓰라(1점).

|  |
| --- |
| void funcA(int a[], int n){  priority\_queue<int> Heap; --------------------------------------------------- ①  for (int I = 0; I < n; i++)  Heap.push(a[i]);  for (int I = 0; I < n; i++) {  a[i] = Heap.top();  Heap.pop();  }  }  int main() {  const int len = 5;  int a[len] = { 5, 3, 6, 8, 9 };  funcA(a, len);  for (int I = 0; I < len; ++i)  cout << a[i];  return 0;  } |

1. 아래 코드(헤더 등 생략) 실행 시 콘솔창 결과를 쓰라(1점).

|  |
| --- |
| int main() {  set<unsigned> dsp;  dsp.insert(1);  dsp.insert(2);  dsp.insert(2);  dsp.insert(3);  for (auto& i : dsp)  cout << i << " ";  return 0;  } |

1. 다음 그래프를 이진탐색트리로 만들려고 한다. 변경이 필요한 경우 그래프를 변경하여 높이가 가장 높은 루트 노드부터, 그리고 왼쪽 노드부터 차례로 노드 키 값을 쓰라. 변경이 필요 없다면 “필요 없다”고 쓰라(1점).
2. 다음 그래프는 AVL 트리인가? AVL 트리인 경우 이유를 쓰고, 아닌 경우 AVL 트리인 이유를 쓰시오. 또한 아래 그래프의 정점을 이용해 AVL 트리를 만들 경우, 최소 높이의 AVL 트리와 최대 높이의 AVL 트리의 노드를 루트 노드부터, 그리고 왼쪽 노드부터 차례로 쓰시오. 그리고 각각의 높이를 쓰시오(1점).
3. 다음 그래프에서 정점 s로부터 시작하여 너비 우선 탐색하여 방문하는 정점을 순서대로 쓰되, 방법이 여럿일 경우 그 중 하나만 택하시오(1점).
4. 다음은 수업 시간에 배운 깊이우선탐색의 수도코드다. 주어진 그래프에서 정점 1을 시작으로 DFS(1)을 수행할 때 ① 라인의 DFS(i)는 몇 번 호출되는가? (s=1 인 경우만 센다.)(1점)

|  |
| --- |
| DFS(s):  set starting point “s”  set parent[s] <- true;  for i adjLists[s].begin() to end()  if (parents[s] != true)  DFS(i) -------------------------------- ① |

1. 다음 그래프를 크루스칼 알고리즘으로 최소 신장 트리가 만들어지는 과정을 쓰시오(1점).

예. {A, H}, {A, D}, …또는 A, D, E, H, … 등

1. 8번에서 주어진 그래프를 프림 알고리즘으로 최소 신장 트리를 구하는 과정을 쓰시오(1점).

예. {A, H}, {A, D}, …또는 A, D, E, H, … 등

1. 다음 그래프에서 정점 s에서 시작하여 다른 정점들에 이르는 최단거리를 다익스트라 알고리즘으로 구하는 과정을 간략히 쓰고, 각 정점까지의 최단 거리를 구하시오(1점).

**Part Ⅱ**

1. [LMS-과제-기말고사] PART Ⅱ 1번 문제를 풀고, FIN-01.cpp 파일을 업로드 하라(3점).
2. [LMS-과제-기말고사] PART Ⅱ 2번 문제를 풀고, FIN-02.cpp 파일을 업로드 하라(3점).
3. [LMS-과제-기말고사] PART Ⅱ 3번 문제를 풀고, FIN-03.cpp 파일을 업로드 하라(3점)
4. [LMS-과제-기말고사] PART Ⅱ 3번 문제에서 ‘탐색 중단’ 문자열은 그래프 이론에서 무엇을 의미하는가?(1점)

**Part Ⅲ**

1. 최대 힙 A[0…n-1]에서 A[0]은 항상 가장 큰 값을 갖고 있다. 반대로 마지막 원소인 A[n-1]은 항상 가장 작은 값을 갖는가?(2점)
2. 다익스트라 알고리즘은 음의 간선이 있는 경우 최단 경로가 보장되지 않는다. 어떤 학생이 다음과 같은 아이디어를 제시했다(2점).

“음의 간선이 있는 그래프에서 모든 간선에 같은 크기의 충분히 큰 상수를 더해 모든 간선을 양수로 만든 다음 다익스트라 알고리즘으로 푼다. 그 결과로 나오는 경로 길이에 포함된(더해진) 상수의 합만큼 빼주면 되지 않겠는가?”

이렇게 하면 최단 경로가 보장되는가? 이유를 설명하시오(2점).

1. 다익스트라 알고리즘은 모든 간선의 가중치가 음이 아닐 때만 사용 가능하다. 음의 가중치를 가진 간선이 있으면 벨만-포드 알고리즘을 사용할 수밖에 없다. 다익스트라 알고리즘의 복잡도는 일반적으로 O(ElogV)이고 벨만-포드 알고리즘의 복잡도는 O(VE)라 큰 차이가 있다. 어떤 그래프에서 단 하나의 간선 가중치만 음수이고, 나머지는 다 양수인 경우 벨만-포드 알고리즘을 쓰기 아쉽다. 이 사실과 해당 음의 간선을 알고 있는 경우, 다익스트라 알고리즘을 두 번 사용하면 작업을 끝낼 수 있다. 첫 번째와 두 번째 다익스트라 알고리즘에 어떤 변화를 주면 되겠는가? 단, 가중치가 음수인 간선을 (x, y)로 하고, 이 음의 간선으로 인해 음의 싸이클이 생성되지 않는다. 편의상 최단 경로의 길이만 구하고 경로 자체를 구할 수 없어도 무방하다. 아이디어를 묻는 문제이니, 알고리즘을 쓰지 말고 간단히 설명할 것)(2점).
2. n개의 정점을 가진 무향 그래프에서 모든 간선의 가중치가 {1, 2, …, n}에 속할 때 여러분이라면 프림 알고리즘과 크루스칼 알고리즘 중 어느 것을 사용하겠는가? 그렇게 답한 이유를 설명하라(2점).
3. 벨만-포드 알고리즘을 이용하면 음의 가중치가 있는 그래프에 대해 각 정점까지 최단 경로를 구할 수 있다. 이 때 정점까지의 최단 경로가 출발지점으로 돌아오는 경로까지 고려하여 각 정점까지의 최단 경로를 구해야 하는 경우가 발생했다. 벨만-포드 알고리즘을 이용한다고 할 때, 여러분이라면 어떻게 해결하겠는가?(2점)