

1. Problem Statement

방향성이 없는 간선에 가중치가 부여된 단순 그래프(simple graph) $G = (V, E)$ 가 주어진다. V 와 E 는 각각 정점 집합과 간선 집합이다. 정점 집합 V 를 두 개의 집합 S 와 $S' (= V - S)$ 로 나누는 방법은 여러 종류가 있는데, 여러분은 S 와 S' 을 잇는 간선의 가중치의 총합을 최대화하도록 나눠야 한다.

그림 1은 10개의 정점과 14개의 간선으로 이뤄진 단순 그래프의 한 인스턴스다. 각 간선의 가중치는 1 또는 -1로 주어져있다. 그림 2는 그림 1의 그래프의 정점 집합을 둘로 나누는 방법 중 하나로, $S = \{1, 6, 8, 9\}$, $S' = \{2, 3, 4, 5, 7, 10\}$ 로 나눈 것이다. S 와 S' 에 속한 정점들을 각각 검은색과 흰색으로 표시하였다. 굵게 표시된 간선은 S 와 S' 을 잇는 간선으로 가중치의 총합은 8이 된다. 그림 3은 정점 집합을 둘로 나누는 다른 방법으로, 두 집합을 잇는 간선의 가중치의 총합은 7이다.

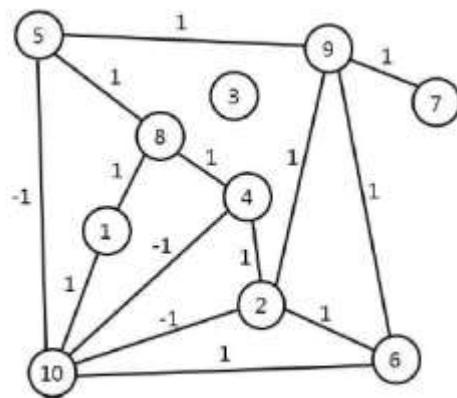


그림 1

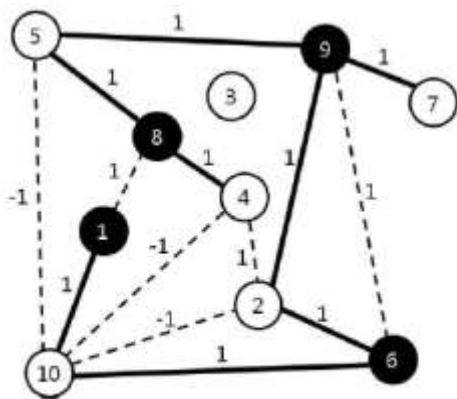


그림 2

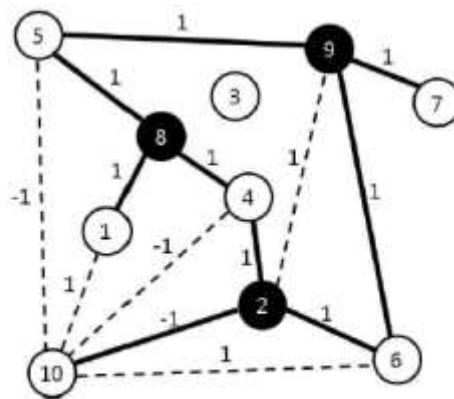


그림 3

주어진 단순 그래프 G 에서 정점 집합을 두 집합 S 와 $S' (= V - S)$ 로 나눌 때, 이 둘을 잇는 간선의 가중치의 총합을 제한 시간 내에 가능한 최대화하는 Genetic Algorithm을 작성하라.

- 인스턴스의 크기: 정점의 개수 $|V| \leq 500$, 간선의 개수 $|E| \leq 5000$

- 제한 시간: 한 인스턴스 당 180초

입력 파일의 이름은 maxcut.in이다. 입력 파일의 각 항목은 공백과 줄바꿈으로 구분된다. 첫 줄에는 정점의 개수 $|V|$ 와 간선의 개수 $|E|$ 가 주어진다. 둘째 줄부터 차례대로 각 간선의 양 끝 정점의 번호와 간선의 가중치가 주어진다. 정점의 번호는 1 이상 $|V|$ 이하의 정수로 주어지고, 간선의 가중치는 정수로 주어진다. 시작점과 끝점이 같은 간선(self-loop)이나 동일한 두 정점을 잇는 복수개의 간선(multiple edges)는 주어지지 않는다.

[출력]

출력 파일의 이름은 maxcut.out이다. 여러분이 구한 최적 집합 S 내의 모든 정점 번호를 한 줄에 공백으로 구분하여 출력한다. 출력 파일에 가중치 총합 수치를 포함하지 않는다.

[입출력 예]

입력 (그림 1의 그래프)

```
10 14
1 10 1
1 8 1
2 10 -1
2 9 1
2 6 1
2 4 1
4 8 1
4 10 -1
5 9 1
5 10 -1
5 8 1
6 10 1
6 9 1
7 9 1
```

출력(그림 2의 $S = \{1, 6, 8, 9\}$)

```
1 6 8 9
```

2. Restriction

이번 과제에서는 순수 GA만 사용하며, 그 중 사용할 수 있는 연산자를 아래와 같이 제한한다.

- Selection: Roulette wheel
- Cross-over: k -point cross-over (e.g. 1-point cross-over, 2-point cross-over, ...)
- Mutation: Typical random mutation
- Replace: any algorithm in the class material

여기서 typical random mutation은 강의에서 설명한 돌연변이를 만드는 방식이다. 순수 GA의 성능과 한계를 체험하기 위해 지역 최적화 등 어떠한 변형 방식도 허용하지 않는다. 이번 과제를 포함, 알고리즘 강좌의 나머지 숙제에서 제출할 프로그램은 반드시 CPU 상에서 single thread로 수행되어야 한다. 멀티코어 혹은 멀티쓰레드 컴퓨팅은 허용하지 않으며, gpu 활용 역시 허용하지 않는다. 프로그램은 외부와 통신 없이 단독으로 수행되어야 한다. 앞의 제약을 위배한 경우 큰 패널티를 받는다.

3. Language

프로그램 언어는 C++, JAVA, 파이썬 중 하나를 사용한다. 시간 제약이 있는 이번 문제의 경우 속도가 빠른 C++, 또는 JAVA를 권장한다. 기본적으로 제공하는 라이브러리 이외의 다른 라이브러리를 사용하는 경우에는 미리 교수와 상의해야 한다. 교수와 상의 없이 다른 언어나 라이브러리를 사용하는 경우, 큰 패널티를 받을 수 있다.

4. Compilation and Execution

제출 시 반드시 Makefile을 포함해야 하며, make all 명령으로 컴파일이 가능하도록 만들어야 한다. C++을 사용하는 경우 컴파일러의 최적화 옵션은 -O3까지 사용할 수 있다. 또한 make run 명령으로 프로그램이 수행되도록 한다. 아래 예시와 같이 명령을 입력했을 때, 여러분이 구한 해가 birthday.out에 기록되어야 한다.

```
$ ls
Makefile yourfile1.cpp yourfile2.cpp ...

$ make all

...
```

```
$ cp /home/alg_Ta/instance1 ./maxcut.in  
  
$ make run  
  
...  
  
$ cat ./maxcut.out  
  
1 6 8 9
```

컴파일과 수행은 리눅스에서 가능해야 한다. 채점 컴퓨터는 Intel Xeon Scalable 4210 이며, OS는 Ubuntu Server 22.04.4 LTS다.

4. Submission

Assignment 1과 동일하다.

4. Team

2024년 알고리즘 수업 과제는 모두 개인 과제다.

5. Report

프로젝트에 대해 보고서를 작성하여 제출한다. 분량은 A4 4페이지 이하다. 보고서는 논문 형식으로 작성한다. 보고서 내에 소스 코드를 그대로 포함해서는 안된다. 포함 시 큰 패널티를 받는다. 보고서 본문에는 최소한 다음 내용을 포함시킨다.

- Genetic algorithm의 correctness와 efficiency
- 해의 표현(chromosome design)
- 사용한 연산자에 대한 설명(selection, crossover, mutation, replacement)
- 함께 제공하는 세 개의 샘플 인스턴스에 대해 GA를 각각 최소 30번씩 수행하여 가장 좋은 결과, 평균 결과, 표준편차를 테이블로 기록(최소 90번의 run)
- 위에서 수행한 GA run 중 하나를 선택, 세대 진행에 따른 population 분석(해들의 평균 품질, 최고 품질)

- Discussion(느낀 점, 잘 안 되는 점, 의외의 현상, 예상대로 된 점 등)

6. Grading

프로그램이 동작할 경우, 보고서와 프로그램이 1:1 비율로 평가한다. 프로그램이 동작하지 않을 경우 평가하지 않는다. 제출한 모든 파일은 점수와 함께 LMS에 공개된다. 지각 제출은 불허하며, 제출 형식을 지키지 않을 경우 0점 처리한다.