

1. 꿀 따기

아래와 같이 좌우로 N 개의 장소가 있다.

9	9	4	1	4	9	9
---	---	---	---	---	---	---

장소들 중 서로 다른 두 곳을 골라서 벌을 한 마리씩 둔다. 또, 다른 한 장소를 골라서 벌통을 둔다. 아래 그림에서 연한 회색의 장소는 벌이 있는 장소이고 진한 회색의 장소는 벌통이 있는 장소이다.

9	9	4	1	4	9	9
---	---	---	---	---	---	---

두 마리 벌은 벌통으로 똑바로 날아가면서 지나가는 모든 칸에서 꿀을 따다. 각 장소에 적힌 숫자는 벌이 지나가면서 꿀을 딸 수 있는 양이다.

1. 두 마리가 모두 지나간 장소에서는 두 마리 모두 표시된 양 만큼의 꿀을 따다. (벌통이 있는 장소에서도 같다.)
2. 벌이 시작한 장소에서는 어떤 벌도 꿀을 딸 수 없다.

위의 그림과 같이 배치된 경우 두 마리의 벌 모두 $4 + 1 + 4 + 9 + 9 = 27$ 의 꿀을 따서, 전체 꿀의 양은 54가 된다.

9	9	4	1	4	9	9
---	---	---	---	---	---	---

위의 그림과 같이 배치된 경우 왼쪽 장소에서 출발한 벌은 $9 + 4 + 4 + 9 + 9 = 35$ 의 꿀을 따고 오른쪽 장소에서 출발한 벌은 $4 + 9 + 9 = 22$ 의 꿀을 따므로, 전체 꿀의 양은 57이 된다.

9	9	4	1	4	9	9
---	---	---	---	---	---	---

위의 그림과 같은 경우는 전체 꿀의 양이 31이 된다.

장소들의 꿀 양을 입력으로 받아 벌들이 딸 수 있는 가능한 최대의 꿀의 양을 계산하는 프로그램을 작성하라.

제약 조건

- $3 \leq N \leq 100\,000$
- 각 장소의 꿀의 양은 1 이상 10 000 이하의 정수이다.

부분문제

1. (11점) $N \leq 20$
2. (13점) $N \leq 500$
3. (31점) $N \leq 5\,000$
4. (45점) 추가적인 제한이 없음.

입력 형식

첫 번째 줄에 장소의 수 N 이 주어진다.

다음 줄에 왼쪽부터 각 장소에서 꿀을 딸 수 있는 양이 공백 하나씩을 사이에 두고 주어진다.

출력 형식

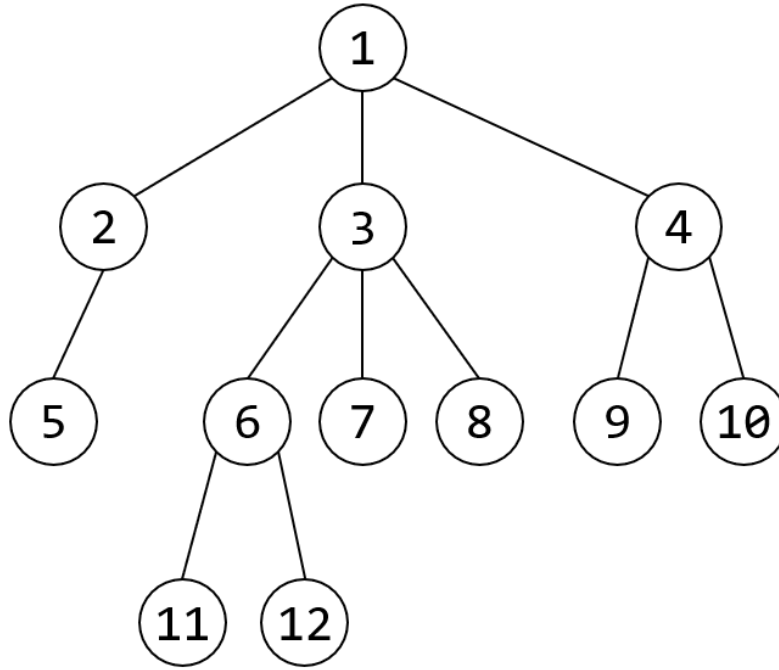
첫 번째 줄에 가능한 최대의 꿀의 양을 출력한다.

예제

표준 입력(stdin)	표준 출력(stdout)
7 9 9 4 1 4 9 9	57
7 9 9 4 1 4 9 9	57
3 2 5 4	10

2. 두 개의 팀

N 명의 사원으로 구성되는 어느 회사의 조직도는 루트 트리(rooted tree)로 표현된다. 트리의 각 노드는 한 명의 사원을 의미하고, 간선은 직속 상사-부하의 관계를 나타낸다. 각 사원은 1부터 N 까지 번호가 부여되어 있다. 사원 1은 회사의 사장이며, 트리의 루트이다. 각 사원의 실력 또한 정수로 표현되는데, 실력은 음수일 수도 있다. 아래 그림은 조직도의 한 예를 보여준다. 노드 안의 수는 사원 번호를 의미한다.



사원 중 일부를 팀장으로 선택하려 한다. 팀장으로 선발되면, 팀장은 자신을 포함하여 팀을 구성해야 하는데, 각 팀은 다음 조건을 만족해야 한다.

1. 팀원은 팀장의 부하 직원이어야 한다. (단, 직속 부하일 필요는 없다.) 예를 들어, 사원 3이 팀장이라면, 사원 8이나 11은 팀원이 될 수 있지만, 사원 1이나 9는 팀원이 될 수 없다.
2. 어떤 사원이 팀원으로 포함되면, 그 사원의 직속 상사도 반드시 같은 팀의 팀원으로 포함되어야 한다 (단, 팀장 제외). 예를 들어, 사원 3이 팀장인 경우, 팀 구성이 {3, 6, 11}인 경우는 가능하나, {3, 8, 11}은 사원 11의 상사인 사원 6이 포함되지 않아 불가능하다.
3. 팀의 점수(팀장을 포함한 팀원 전체의 실력의 합으로 정의)가 최대가 되도록 팀원을 구성해야 한다. (팀의 점수가 최대가 되도록 하는 팀 구성이 유일하지 않으면, 그 중 아무거나 골라도 된다.) 예를 들어, 모든 사원의 실력이 모두 양수라고 가정했을 때, 사원 3을 팀장으로 선발하면, 팀 구성은 반드시 {3, 6, 7, 8, 11, 12}가 되어야 되고, 이 중 한명이라도 팀원에서 빠지는 경우는 허용되지 않는다.
4. 팀원은 다른 팀의 팀원(팀장 포함)이 될 수 없다. 따라서 어떤 사원 A가 팀장일 때, 부하직원 B가 조건 3에 의해 팀장 A의 팀원이 되어야 한다면, 사원 A와 B를 모두 팀장으로 선발하는 것은 불가능하다.

회사에서는 두 명의 팀장을 선택하여 두 개의 팀을 구성하되, **두 팀의 점수를 합산**한 결과(즉, 두 팀의 팀원 전체의 실력의 합)가 최대가 되도록 하려고 한다. 두 팀 점수의 합으로 가능한 최댓값을 계산하는 프로그램을 작성하라. 위 조건을 만족하도록 두 팀을 구성할 수 있는 경우만 입력으로 주어진다.

제약 조건

- $2 \leq N \leq 200\,000$
- 사원의 실력은 $-1\,000\,000\,000$ 이상, $1\,000\,000\,000$ 이하의 정수이다.
- 사원 1을 제외한 각 사원 i 의 직속 상사 번호는 $i - 1$ 이하이다.
- 지문에 제시된 조건을 만족하도록 두 팀을 구성할 수 있다.

부분문제

1. (17점) 모든 사원의 실력은 양수이다.
2. (12점) $N \leq 5\,000$ 이고, 사원 1을 제외한 각 사원 i 의 직속 상사 번호는 $i - 1$ 이다.
3. (20점) 사원 1을 제외한 각 사원 i 의 직속 상사 번호는 $i - 1$ 이다.
4. (16점) $N \leq 400$.
5. (17점) $N \leq 5\,000$.
6. (18점) 추가 제약 조건 없음

입력 형식

첫 번째 줄에 하나의 정수 N 이 주어진다.

다음 N 개의 줄에는 사원들에 대한 정보가 주어진다. 이 중 i ($1 \leq i \leq N$)번째 줄에는 사원 i 의 실력과 직속 상사 번호를 나타내는 두 개의 정수가 공백 하나를 사이로 두고 주어진다.

사원 1은 직속 상사가 없으므로, 사원 1의 직속 상사 번호의 자리에는 대신 -1 이 들어온다.

출력 형식

두 팀 점수의 합으로 가능한 최댓값을 출력하라. 출력 값이 매우 클 수 있으므로 C, C++ 언어에서는 long long 형의 변수를, Java에서는 long 형의 변수를 사용해야 한다.

예제

표준 입력(stdin)	표준 출력(stdout)
4 3 -1 -2 1 2 2 -1 1	5