

[railroad]

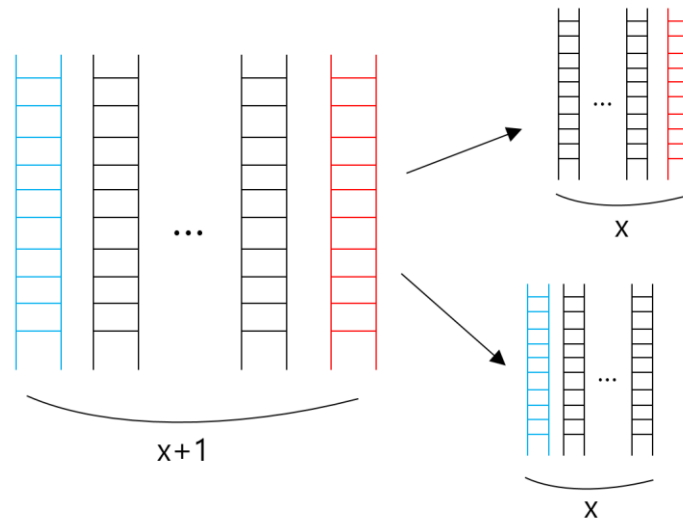
먼저 나머지 연산을 따로 고려하지 않기 위해서 987654321에 해당하는 값을 매크로 정의하고, 덧셈과 곱셈을 수업시간에 했던 것처럼 미리 함수로 정의하고 시작하였습니다.

문제를 해결하는 과정에서 처음에는  $N=1$ 인 상황을 먼저 고려했습니다. 즉, 다른 역은 신경 쓰지 않고, 철도만  $x$ 개 있는 상황에서 몇가지 경우의 수가 나올지를 먼저 고려했습니다.

$x=1$  일 때, 1

$x=2$  일 때, 2

$x=3$  일 때, 4까지는 손으로 계산을 하였고, 그 다음은 점화식을 세워서 접근하였습니다.



그림으로 묘사하면 위와 같습니다. 철도의 개수가  $x+1$ 일 때는 파란색 철도나 빨간색 철도 중 하나를 택해야 하고, 그 이후에는 철도의 개수가  $x$ 개인 상황이 됩니다.

즉, 철도의 개수가  $x$ 개일 때 경우의 수를  $a_x$ 라고 하면  $a_{x+1}=2a_x$ 라는 관계를 찾을 수 있습니다.

따라서  $a_x=2^{x-1}$ 이라는 것을 알 수 있습니다.

그 다음은  $N$ 개의 역을 고려하였습니다.

각  $N$ 개의 역에서 각각의  $a_x$ 개의 공사를 할 수 있는 경우를 택할 수 있고,  $x$ 개의 공사를 해야하는 상황입니다.

즉, 총  $\prod a_{x_i}$ 의 경우를 골라서,  $\sum x_i$ 번의 공사를 진행해야 하는 상황으로 생각할 수 있습니다.

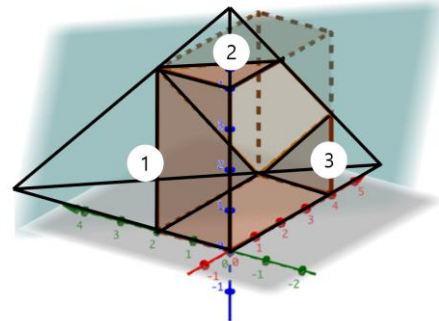
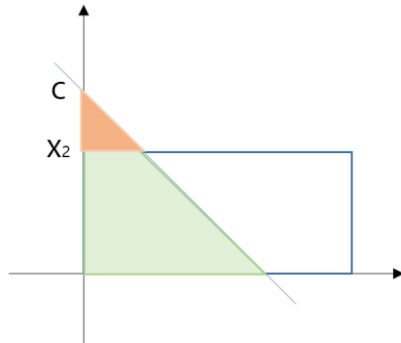
이를 수식으로 쓰면,

$$\left(\sum_{x_1}^{x_i}\right)\left(\sum_{x_2}^{x_i-x_1}\right)\dots\left(\sum_{x_n}^{x_n}\right)\times\prod 2^{x_i-1}$$

Combination의 경우에는  $\sum x_i$ 의 최댓값이 5000이므로 table을 만들어 그 값들을 모두 찾아주었고, power의 경우에는, 재귀적으로 구현하였습니다.

[cake]

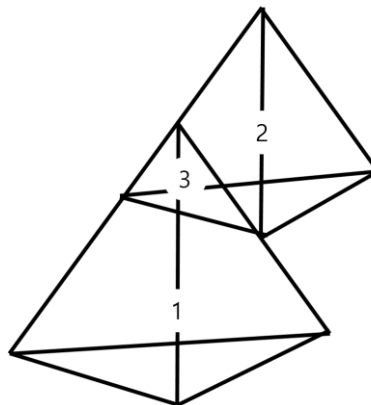
위 문제와 마찬가지로 987654321에 해당하는 값을 매크로로 지정하고, 덧셈, 곱셈, 뺄셈을 수업시간에 했던 것처럼 미리 함수로 구현하였습니다.



우선 시각화가 가능한 2차원과 3차원 상황을 먼저 고려하였습니다. 내부의 사각형을 축에 평행하게 쪼개서 더하거나, 삼각형으로 나눠서 더하는 방식을 먼저 생각하였으나, 이를 일반화하여 다차원으로 확장시키기에 무리가 있었습니다.

그 다음으로 떠올린 방법은 포함-배제의 원리입니다. 왼쪽 2차원의 경우에는  $c^2/2$  넓이의 녹색 삼각형에서  $(c-x_2)^2/2$  넓이의 주황색 삼각형을 뺀 값이 넓이가 되고, 오른쪽 3차원의 경우에는 전체의  $c^3/3!$ 의 삼각뿔에서 1, 2, 3번의 삼각뿔의 부피를 빼준 것이 부피가 됩니다.

빼줘야 하는 도형의 한 변의 길이는  $x_1+x_2+\dots+x_n=c$  에서  $c$ 에서  $x_i$  를 빼준 값이 됩니다. 이 때 그 값이 음수가 된다면 왼쪽 도형에서 보는 것처럼 고려해줄 필요가 없습니다.



그러나, 한번의 길이가 충분히 길어지게 되면 빼주는 삼각뿔들이 중복되는 경우가 생깁니다. 이런 경우에는 그 값을 다시 더해줘야 하는데, 그 더하는 삼각뿔의 한 변의 길이는 1번과 2번에서 빼준  $x_i$ 을 한번에 빼주면 구할 수 있습니다. 이 값이 음수라면 실제로는 중복되지 않는 상황인 것입니다.

따라서  $x_1, x_2, \dots, x_n$  의 모든 경우를 다 따져봐야 했고, 각각을 이진수 자릿수로 생각하여  $2^n$ 가지의 경우를 모두 체크하였습니다.

실제 구현에서는 이전문제에서 사용하였던 power 함수를 사용하였습니다.