

Wq5

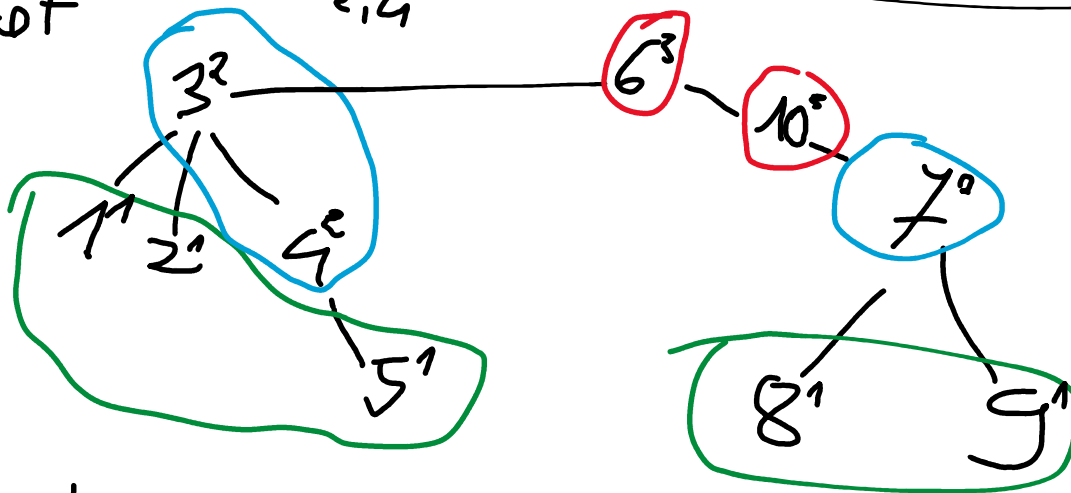
Wejście		
5 4		4
0 1 -10 5		
1 1 -10 5		
1 0 -10 2		
0 0 -1 0		
-1 -1 0 1		

0 0 0 0 0 0 0  
 0 1 1  
 0 0 0 0 0 0 0

$$dp[i][j] = \max(dp[i-1][j], dp[i][j-1]) + value[i][j]$$

Fot

2,4



$\left\lfloor \frac{k}{2} \right\rfloor$  warstw (+1 fotowada, jeśli  $2 \nmid k$ )

Bfs od liści

Flp

## Przykład

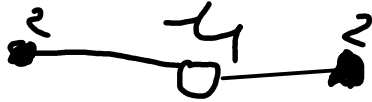
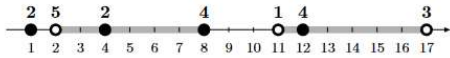
Dla danych wejściowych:

17  
2 -5 0 2 0 0 4 0 0 -1 4 0 0 0 -3

poprawnym wynikiem jest:

12

**Wyjaśnienie przykładu:** Poniżej zilustrowano test przykładowy zawierający  $n = 17$  miast, w których zostaną wybudowane trzy fabryki (białe kółka) i cztery elektrownie (czarne kółka). Zaznaczono także poprawną sieć energetyczną o długości 12 kilobajtometrów (szare odcinki).



```
for (i = 1; i < n; i++)
    for (j = 0; j < i; j++)
        if (sum[i] - sum[i - j - 1] > 0)
            dp[i] = min(dp[i], dp[i - j - 1] + j)
```

$O(n^2)$

$\boxed{\text{sum}[i]} \geq \text{sum}[i - j - 1]$

query (qpos (sum, i))

sum[i] - kłopot z wejściami

sorted\_s[i] - sorted (sum, i)

```
for (i = 1; i < n; i++)
```

```

for (i = 1; i <= n; i++)
    auto it = lower_bound (sum[i], i);
    int pos = it - sum.begin();
    dp[i] = t.query (0, pos) + time
    ++time
    t.update (pos, dp[i] - time)

```