Оптимизации на динамични

Divide and conquer optimization

1. Обща формула: $dp[n][k] = min/max(dp[i-x][k-1] + cost(i,n))$.
Най-често x е равно на 0 или 1.

2. Условие за оптимизацията: $opt[1][k] \le opt[2][k] \le \cdots \le opt[n][k]$,
където $opt[n][k]$ е най-малкото $x \le i < n$, за което е изпълнено:

$$dp[n][k] = dp[i-x][k-1] + cost(i,n))$$

3. Достатъчно (но не необходимо) условие изразено с $cost()$:
Нека $1 \le a \le b \le c \le d \le N$
при минимизация: $cost(a,c) + cost(b,d) \le cost(a,d) + cost(b,c)$
при максимизация: $cost(a,c) + cost(b,d) \ge cost(a,d) + cost(b,c)$

Knuth's optimization

1. Обща формула: $dp[i][j] = min/max(dp[i][k-x] + dp[k][j]) + cost(i,j)$. Най-често х е равно на 0 или 1.

2. Условие за оптимизацията: $opt[i][j-1] \le opt[i][j] \le opt[i+1][j]$, където $opt[i][j]$ е най-малкото $x + i < k < j$, за което е изпълнено:

$$dp[i][j] = dp[i][k-x] + dp[k][j] + cost(i,j)$$

3. Достатъчно (но не необходимо) условие изразено с $cost()$:

   Нека $1 \le a \le b \le c \le d \le N$

   при минимизация: $cost(a,c) + cost(b,d) \le cost(a,d) + cost(b,c)$

   при максимизация: $cost(a,c) + cost(b,d) \ge cost(a,d) + cost(b,c)$

   и при двете: $cost(b,c) \le cost(a,d)$ (монотонност)

1D1D optimization

1. Обща формула: $dp[n] = min/max(dp[i-x] + cost(i,n))$. Най-често x е равно на 0 или 1. Нека $dp2[i][j] = min/max(dp[k-x] + cost(k,i))$ за $k \le j$.

2. Условие за оптимизацията: $opt[i][j-1] \le opt[i][j] \le opt[i+1][j]$, където $opt[i][j]$ е най-малкото $x \le k \le j$, за което е изпълнено:

$$dp2[i][j] = dp[k-x] + cost(k,i)$$

Оттук следва, че $dp[n] = dp[opt[n][n-1] - x] + cost(opt[n][n-1], n)$

3. Достатъчно (но не необходимо) условие изразено с $cost()$ – същото като при divide and conquer optimization.

Източници

- https://codeforces.com/blog/entry/86306
- https://codeforces.com/blog/entry/93772
- https://codeforces.com/blog/entry/8219