Dawid Pawliczek
Lista 2, Zadanie 2

Treść zadania

Danych jest n odcinków $I_j = \langle p_j, k_j \rangle$ leżących na osi OX, $j = 1, \dots, n$. Ułożyć algorytm znajdujący zbiór $S \subseteq \{I_1, \dots, I_n\}$ nieprzecinających się odcinków o największej mocy $|S|$.

Intuicja

Dynamiczne DP (pomysł odrzucony). Można próbować rozwiązać problem przez DP: $\mathrm{dp}[i][j]$ – maksymalna liczba odcinków mieszczących się w przedziale $[i, j]$. Wzór rekurencyjny:

$$\mathrm{dp}[i][j] = \max_{\substack{I=\langle k,k'\rangle \\ i\le k<k'\le j}} \bigl(\mathrm{dp}[i][k] + 1 + \mathrm{dp}[k'][j]\bigr),$$

jednak prowadzi to do kosztu kwadratowego.

Kluczowa obserwacja (wymiana). W *dowolnym* optymalnym rozwiązaniu można wybrać odcinek, który kończy się *najwcześniej* spośród wszystkich. Jeśli optymalne rozwiązanie tego nie robi, zamieniamy jego pierwszy odcinek na odcinek o najmniejszym k: liczność zbioru nie spada, więc otrzymujemy równorzędne optimum.

Stąd wynika zachłanna konstrukcja: wybieramy najwcześniej kończący się odcinek, wyrzucamy wszystkie z nim kolidujące i powtarzamy na pozostałych.

Algorytm zachłanny

Algorithm 1 BestIntervals

Require: zbiór odcinków $I = \{I_j = \langle p_j, k_j \rangle\}$
Ensure: maksymalny zbiór rozłącznych odcinków

posortuj odcinki niemalejąco po k_j
$S \leftarrow []$; $\quad \mathit{lastEnd} \leftarrow -\infty$
for all $I_j = \langle p_j, k_j \rangle$ w tej kolejności **do**
    if $p_j \ge \mathit{lastEnd}$ **then**
        dołącz I_j do S
        $\mathit{lastEnd} \leftarrow k_j$
    end if
end for
return S

Dowód poprawności

Dowód przez *wymianę*. Niech S_g – zbiór zwrócony przez algorytm, $S^\star$ – dowolne optymalne rozwiązanie, a odcinki obu zbiorów posortowane rosnąco po końcach: $S_g = (g_1, g_2, \dots)$, $S^\star = (s_1, s_2, \dots)$.

1. Pierwszy odcinek algorytmu, g_1, ma najmniejszy koniec spośród wszystkich, więc $k(g_1) \le k(s_1)$. Jeżeli $g_1 \ne s_1$, konstruujemy zbiór $S' = (g_1, s_2, s_3, \dots)$. Ponieważ $p(s_2) \ge k(s_1) \ge k(g_1)$, g_1 nie przecina s_2, zaś liczność $|S'| = |S^\star|$, więc S' także jest optymalny.

2. Stosując tę samą operację wymiany indukcyjnie dla kolejnych pozycji, otrzymujemy optymalne rozwiązanie, które zaczyna się prefiksem $g_1, g_2, \ldots$. Po skończeniu indukcji całe S_g musi być optymalne, czyli $|S_g| = |S^\star|$.

$\square$

Złożoność

Sortowanie – $O(n \log n)$, pętla – $O(n)$. Łącznie $O(n \log n)$. Pamięć pomocnicza $O(1)$ (poza tablicą wejściową).