Zadanie: PRT
Program telewizyjny

XI obóz informatyczny, grupa początkująca, dzień 1. Dostępna pamięć: 32 MB. *25.02.2012*

Rozwiązanie wzorcowe $O(n * log^2 n)$

Zauważmy, że jeżeli da się skonstruować rozwiązanie z wynikiem większym bądź równym r, to równie dobrze możemy skonstruować rozwiązanie o wyniku p, gdzie $p \leq r$. Dodatkowo, jeżeli nie da się skonstruować rozwiązania o wyniku większym bądź równym r, to też nie da się skonstruować rozwiązania o wyniku p, gdzie $p \geq r$. Wykorzystując te dwa fakty, możemy binarnie wyszukać wynik.

Podczas wyszukiwania binarnego, gdy sprawdzamy wartość r, musimy sprawdzić czy da się tak wybrać co najmniej k programów o atrakcyjności $a_i \geq r$, tak aby było się nie pokrywały. Do tego możemy zastosować prosty algorytm wykorzystujący programowanie dynamiczne.

Jeżeli nie ma co najmniej k programów spełniających $a_i \geq r$ oczywistym jest, że dla tego r nie da się skonstruować rozwiązania. Najpierw posortujmy zbiór programów o atrakcyjności $a_i \geq r$ rosnąco po e_i. Ciąg posortowanych tak indeksów programów nazwijmy m_j. Niech $dp[x]$ będzie równe największej możliwej liczbie programów, które można wybrać spośród pierwszy x programów, tak aby się nie pokrywały. Budujemy tą tablicę według wzoru:

$$dp[x] = \begin{cases} 1 & \text{gdy } x = 1 \\ max_{1 \leq y < x \wedge e_{m_y} < b_{m_x}}(dp[y]) + 1 & \text{gdy } x > 1 \end{cases}$$

Gdybyśmy brutalnie wypełniali tę tablicę otrzymalibyśmy złożoność $O(n^2 * logn)$. Jednak warto zauważyć, że poszukiwany w powyższym równaniu y możemy wyszukać binarnie w posortowanej liście programów. Wykorzystując to, złożoność zmniejsza się do $O(n * log^2 n)$, która nas satysfakcjonuje.