Авторски анализ на задача **jumps**

Решение за 3 точки

Първата подзадача я дадох не толкова за състезателите, колкото за да могат малките деца (6-7 клас) да се упражняват да пишат рекурсия. Няма много какво да се коментира по нея.

Решение за 13 точки

Тук отново нямаме нищо особено интересно - трябва да осъзнаем, че рекурсивният ни процес ще се оптимизира многократно с меморизация.

Решение за 20/35 точки

Главната идея тук е, че е малко безсмислено са мислим за поредните скокове с еднаква дължина като за отделни събития, ако между тях няма “почивка”. Ако ги обединим в един “блок” получаваме следната задача: мога да скоча от позиция i до позиция j и да получа $d \times A_j$ точки, където d е делител на $j - i$. Различните резултати се получават при различни по скорост реализации на тази идея.

Решение за $A_i > 0$ случая

Тук зарових кучето.

Ако почнем да си играем с един пример, в който всяко число е положително, бързо трябва да видим, че всъщност не са ни нужни особено големи скокове - скок с дължина 4 още от миналата подзадача трябва да е ясно, че е ненужен. Тук интересно за разглеждане е, че скокове с дължина 5 също са ненужни - да кажем скочил и съм взел A_i пари. Еми наместо това можеше да скоча веднъж с дължина 2, да взема A_{i-3}, и после пак да скоча до A_i което исках да взема. Тъй като всички числа, в частност A_{i-3}, са положителни това видимо е по-добрият вариант. Този аргумент се разширява и за по-големите скокове и стигаме до следното наблюдение.

Единствените скокове*, които ще ползваме са с дължина* 2 *и* 3*.*

Дори да го видим това обаче забавата от задачата не е приключила! Нека си представим рекурентната зависимост(което впрочем винаги е добра идея при дп задачи).

$$dp_i = max_{j<i}(dp_j + (i - j)/k \times A_i),\ k = 2 \text{ или } 3.$$

Това може би изглежда малко сложно, на по-опитни състезатели веднага ще им заприлича на *Convex hull trick.* И наистина, ако направим малко преобразувания…

$$dp_i = max_{j<i}(dp_j + i/k \times A_i - j/k \times A_i)$$

$$dp_i = max_{j<i}(dp_j - j/k \times A_i) + i/k \times A_i$$

Получихме точно това, което ще очакваме от една *Convex hull trick* задача.
Не на последно място, трябва да внимаваме и как точно свързваме позициите - не можем да свържем четна и нечетна позиция със скокове от дължина 2. Тоест като почнем да скачаме с дължина k и сме тръгнали от $i \equiv r(mod\ k)$ трябва да се приземим на позиция $j \equiv r(mod\ k)$. За да се погрижим за това авторовото решение ползва масив от *CHT small[3][3].*

Пълно решение

Както обещах, пълното решение ще е продължение на миналото. За удобство тук ще разгледаме няколко случая:

1. приземявам се на $A_i < 0$

   това е по-малкият от двата случая, заради това него гледаме първи - нямам никакъв интерес да умножавам отрицателно число, заради това ще потърся най-оптималната стартова позиция - тази с максимално dp_i и ще скоча от нея. Трябва да внимавам да не се окаже, че правя скок с дължина 1 - тези са забранени по условие.

2. приземявам се на $A_i > 0$

   Ако не сме забелязали досега, вече е време да осъзнаем, че няма смисъл да правим скокове с дължина съставно число (тук да не се бърка с “блок”). Така че разглеждаме единствено скокове с дължина k просто число. Те пък отново се разделят на два случая:

   - ако $k > \sqrt{N}$: такива скокове ще правим “самички”, тоест няма нужда да правим два поредни скока с дължина над корена. Ще докажа това в следващия случай.
   - ако $k \leq \sqrt{N}$: тук просто ще разширим решението ни от подзадачата с $A_i > 0$. Причината да разглеждам сксковете спрямо $\sqrt{N}$ е следната:

     $$\text{блок} = \text{дължина на скока} \times \text{брой на скокове}$$

     Тоест няма как хем да са дълги скоковете, хем да са много - даже видяхме ако скоковете са “дълги” няма смисъл да правим по няколко такива заедно.

Интересно е да помислим за сложността на алгоритъма ни, която е $O(N(\sqrt{N}/log(\sqrt{N}))log(\sqrt{N}))$. С други думи прости $O(N\sqrt{N})$ - логаритъмът от двоичното търсене в *Convex hull trick* се взима от размера му, който ще стане максимум $\sqrt{N}$ (запомнете, че в зависимост от остатъците си различните позиции влизат в различни *CHT-ma*). Обаче пък има смисъл да работя само с простите числа по-малки от $\sqrt{N}$, а добра оценка на броя прости числа по-малки от $\sqrt{N}$ е $\sqrt{N}/log(\sqrt{N})$.

П.П1: След като решите тази задача, може да ви се стори интересно да решите и тази https://codeforces.com/contest/1207/problem/F
П.П2: Заглавието на задачата е препратка към песента Rich Girl на Gwen Stefani.

Иван Лупов