НОИ1 B, 2026, Spiderman

Подгрупа 1:

Очевидно е че с ограничение N<=20 ще използваме метод за решение чрез пълно изчерпване (или bitmask dp, но не в случая). Обхождаме всички числа mask от $[0;2^n-1]$, където за всяка единица в двоичния запис означава, че искаме сградата да участва в тренировката. Попълваме в един вектор височината и номера на съответните сгради. Сортираме ги по ред на големина и правим проверка дали текущата комбинация е валидна. Сложност $O(2^n*n^2)$

//spiderman_2^n.cpp

Има два главни подхода за продължение на задачата, силно ви препоръчвам първия, защото е доста по-кратък за писане и разбиране.

I ВАРИАНТ

Движим се от по-ниските към по-високите блокове. Използва се конструирането на граф от дадения масив, където се прави ребро наляво и надясно на първата сграда по-висока от текущата (със стек). След което се пуска dfs за пресмятане на максималната дълбочина от ребра, тоест за най-дълга тренировка от подскачане на сгради. Сложност: O(n)

//author.cpp

II ВАРИАНТ

Движим се от по-високите към по ниските блокове.В случая ще означим с dp [l][r], което да пази за интервал от масива [l;r], от колко сгради ще се състои отговорът. Започваме като първоначално гледаме кои са най-големите числа в масива. Ясно е, че ако спайдъмен скочи в някой блок наляво, по-нисък от текущия – той няма да може да се върне надясно(и обратно). Правим си рекурсивна функция която смята отговор на интервал [l;r]. Затова за всеки от най-високите блокове от l до r *(може да има няколко най-високи)* означаваме с dp кой отговор ще е по-добре: да скочим в левия му интервал, или в десния му интервал.

И съответно си служим със сегментно дърво за намиране на максимум в интервал (log(n)). След всяко такова намиране на максимум в интервал заменяме стойността му в интервалното с 0 и продължаваме с алгоритъма. Сложност: O(n*log(n)), Памет: O(n), достъпът се улеснява, чрез структурата map, която помага за по-лесно организиране на данните по компактен начин, ако сме сигурни че броят на интервалите които викаме са в допустимото ограничение. В нашият случай за всяка сграда имаме само един интервал [l;r], за който търсим отговора.

//spiderman_segment_tree.cpp

Задачата е взета от системата CSES -> Mountain Range където я няма със събтаскове.