АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА САЛАМ

Задачата се решава използвайки метода на динамичното оптимиране. Нека имаме масив best_to[] (0-based), в който в i-тата клетка ще пазим възможно най-голямата наслада, която Иван може да изпита, ако сантиметрите от 1 до i+1 (включително) са на разположение само на него (т.е. Петър ще изяде максимум до i+2-вия сантиметър). Нека в масива a[] (0-based) сме запомнили вкусовите качества на всеки сантиметър на салама.

best_to[0] = max(0, a[0])

best_to[i] = max(best_to[i-1], sum(a[0], a[1], …,a[i])

За целите на втората част от задачата трябва да пазим и до кой сантиметър се получава тази най-голяма наслада – това се случва в масива best_to_left[] (0-based), като когато го смятаме трябва да взимаме предвид изискването за възможно най-голяма част изядена от Иван, т.е. ако sum(a[0], a[1], …, a[i]) == best_to[i-1], за нас е по-добро решение да „изядем“ всички сантиметри от 0 до i, отколкото да вземем еквивалентното (по наслада) решение до i-1.

Веднъж имайки сметнат best_to[], започваме да пробваме Петър да изяде 0 сантиметра от салама, после 1 сантиметър, после 2 и т.н. – т.е. изчерпваме всички възможности за Петър. Ако Петър изяде j сантиметра от салама, то в този случай най-доброто решение е best_to[n-(j+1)] + sum(a[n-1], a[n-2], …, a[n-j]. От всички тези варианти трябва да изберем този, който е оптимален за нас, а именно: този с най-голяма наслада. Ако има много такива, този в който best_to_left е максимален. Ако има много такива, този в който Петър изяжда най-голяма част също – т.е. за най-голямо j.

Автор: Момчил Иванов