

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ВОЙНА

Преведено условието на „информатически“ език, то гласи: дадено ни е дърво, да се избере такъв корен на това дърво, така че разликата в дължините на пътищата от този корен до най-отдалеченото листо и до най-близкото листо е минимална.

Решението за 50 точки се изразява в това, да бъде избран всеки възможен връх за корен, да бъдат сметнати дължините на пътищата до листата (което става с обхождане в дълбочина или обхождане в ширина) и да бъде избран този, за когото гореспоменатата разлика е минимална. Сложността на такова решение е  $O(N^2)$

Решението за 100 точки е със сложност  $O(N)$ . Целта ни е за всеки връх да намерим разстоянието до най-близкото листо и разстоянието до най-далечното листо. За простота на обясненията ще търсим само разстоянието до най-близкото листо – другото разстояние се смята абсолютно аналогично и успоредно.

Нека си изберем произволен връх  $r$ , който за целите на нашето решение ще използваме за корен на дървото. В решението на автора този връх не е съвсем произволен, а е задължително със степен поне 2, но това е необходимо само за целите на по-изчистена реализация. Решението е на две фази: първо обхождаме дървото от листата към корена  $r$  и на втората фаза разглеждаме дървото от корена  $r$  към листата.

Целта на първото обхождане е да намерим за всяко поддърво с корен  $r'$  най-близкото до корена  $r'$  листо (и дължината на пътя до него) в рамките само на това поддърво. Всъщност търсим и нещо друго: ако най-близкото листо до  $r'$  е в поддърво с корен  $v$ , където  $v$  очевидно е наследник на  $r'$ , то търсим и най-близкото листо, което, обаче, е в друго поддърво (т.е. поддърво с корен някой друг наследник на  $r'$ ). По-късно ще разберете защо ни е нужно това. Изчисляването на „първа фаза“ авторът изчислява с обхождане в дълбочина. Това съвсем успешно може да бъде реализирано и с обхождане в ширина, като се започне от листата и се продължи нагоре по дървото към корена  $r$ . Нека текущо разглеждания връх в обхождането е  $c$ . Рекурентната зависимост, с която се смятат търсените две разстояния е следната:

$$\min_1[c] = \min(\min_1[v] + d(c, v)), \text{ където:}$$

$\min_1[]$  е разстоянието до най-близкото листо от даден връх,

$v$  е някой пряк наследник на  $c$ ,

$d(c, v)$  е дължината на реброто  $(c, v)$ .

Нека този минимум се постига в някое  $v'$ , тогава:

$$\min_2[c] = \min(\min_1[v] + d(c, v)), \text{ където:}$$

$\min_2[]$  е разстоянието до най-близкото листо от даден връх, използвайки различни поддърво от това с най-близкото листо (което поддърво се използва за  $\min_1$ ).

$v$  е някой пряк наследник на  $c$ , **различен от  $v'$** ,

$d(c, v)$  е дължината на реброто  $(c, v)$ .

Втората фаза се състои в обхождане на дървото от корена към листата (т.е. чрез обхождане в дълбочина) и смисъла на това обхождане е за всеки връх да бъде сметнат вече истинския минимум като разстояние до някое от листата на дървото (а не само до тези които са наследници на върха). Идеята е когато сме във връх  $c$  и ни предстои рекурсивно да обходим връх  $v$ , който очевидно е пряк наследник на  $c$ , да „подадем“ на  $v$  информация за най-близкото листо, което не му е наследник, и щом не му е наследник, значи задължително пътя до него преминава през непосредствения му предшественик, именно върха  $c$ . Така обхождането в дълбочина има два параметъра:

$c$  – текущо обхождания връх и  
 $upperMin$  – разстоянието до най-близкото листо, до което стигаме задължително преминавайки през непосредствения предшественик на  $c$ .

В началото на обхождането на всеки връх лесно можем да сметнем разстоянието до най-близкото листо в цялото дърво до текущо разглеждания връх  $c$ , и той е именно:

$$\min(upperMin, \min1[c])$$

И така, нека сме във връх  $c$  и ни предстои да обходим непосредствения му наследник  $v$ . Това което трябва да му подадем е:

$\min(upperMin, \min1[c]) + d(c, v)$ , когато листото в  $\min1[c]$  не е в поддървото с корен  $v$ , или

$\min(upperMin, \min2[c]) + d(c, v)$ , когато листото в  $\min1[c]$  е в поддървото с корен  $v$ .

Както беше споменато по-горе, нищо не пречи използвайки същите разсъждения успоредно с намирането на разстоянието до най-близкото листо за всеки връх да сметнем и разстоянието до най-далечното листо. Веднъж имайки тези две за всеки връх, просто обхождаме всички и намиране този, за който разликата между двете е най-малка. Така задачата е решена със сложност  $O(N)$ . За да се избегнат евентуални проблеми със `stack overflow`, авторът е предложил решение с итеративна имплементация на обхождането в дълбочина, използвайки стек.

Автор: Момчил Иванов