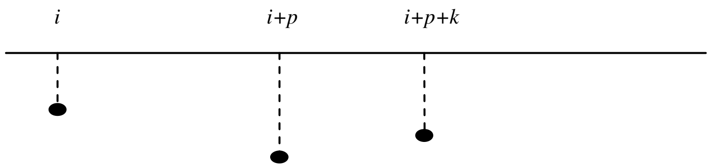
АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
ПОДВОДНИЦИ

Наивното решение на задачата е да се поддържа масив с дълбочините на позициите, на които плуват всички подводници, който се актуализира при всяка команда от тип „размени позиции“. При команда „изпрати сигнал“ се преброява колко сигнала ще получи подводницата на всяка позиция и се извежда най-големият намерен брой. За целта, за всяка подводница се намира коя е подводницата, която ще получи нейния сигнал, т.е. най-близката подводница, която плува след нея на по-голяма дълбочина и за тази подводница се добавя единица в брояча на сигналите, които получава. Такова едно преброяване за всички подводници е със сложност $O(N^2)$, а цялото решение е със сложност $O(N^2*M)$ в най-лошия случай. Такъв алгоритъм решава подзадача 1.

Важно съображение, което ще опрости реализацията на наивното решение, а за следващите решения е задължително, е следното:

При зададеното разстояние от 5км между две съседни позиции в бойния ред и максималната дълбочина от 3 км, на която плуват подводниците, най-близката по евклидово разстояние до дадена подводница, която плува на по-голяма дълбочина след нея, е тази, която се намира на най-близка позиция след нея в бойния ред и плува на по-голяма дълбочина.



Наистина нека имаме подводници, които плуват на позиции i, $i+p$ и $i+p+k$. Тогава най-голямото евклидово разстояние в километри между подводниците на позиции i и $i+p$ е по-малко от $\sqrt{25p^2+9}$, а най-малкото разстояние между подводниците на позиции i и $i+p+k$ е по-голямо от $5(p+k)= \sqrt{25(p+k)^2} > \sqrt{25p^2+9}$.

Това съображение ни дава възможност да не се интересуваме от евклидовото разстояние между подводниците, а само от подредбата им в бойния ред и от дълбочината, на която плуват.

Първата стъпка към подобряване на решението е да направим определянето на максималния брой сигнали, получавани от една подводница при една команда „изпрати сигнал“, линейно по N.

Решение със сложност $O(N*M)$

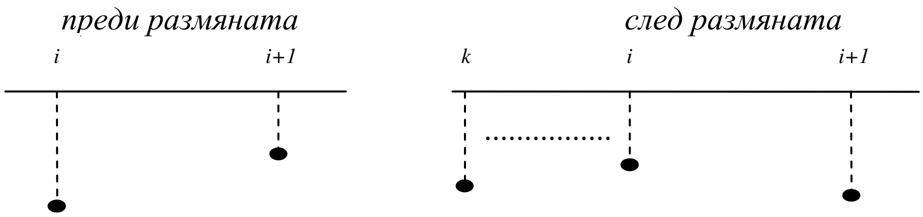
Определянето на броя сигнали, които получава всяка подводница след изпълнението на команда „изпрати сигнал“ (и съответно на най-големия брой) може да се извърши с едно преминаване през масива с дълбочините на позициите на подводниците, като се използва стек. В стека се намират дълбочините на онези позиции, на които плуват подводници, за които все още не е ясно коя подводница ще получи сигнала им. Разглеждайки поредната подводница от реда, ние вадим от стека и броим онези подводници, чийто дълбочина е по-малка от дълбочината на разглежданата подводница. Сигналът на всяка от тях ще бъде получен от разглежданата подводница (фактът, че се намират все още в стека говори, че не е имало подводница, която е след тях в бойния ред, плува по-дълбоко от тях и се намира по-близко до тях от разглежданата). След като преброим сигналите, получавани от разглежданата подводница, ние сравняваме техния брой с намерения до момента максимален брой и, ако е по-голям, обновяваме текущия максимум. След това вкарваме дълбочината на разглежданата подводница в стека. Реализацията на тази операция може да се види във функция **calc_sub_max** в програма **submarines2**. Ясно е, че при M команди сложността на това решение в най-лошия случай е $O(N*M)$. То е реализирано с програма **submarines2** и решава подзадача 2.

Решение със сложност $O(max(N,M*(logN)^2)))$

Решението на подзадача 3 е базирано на идеята, че след като веднаж, в началото, сме изчислили чрез стек коя подводница колко сигнала получава, можем значително по-бързо да преизчисляваме коя подводница колко сигнала би получила след всяка команда „размени позиции“. Съображението е, че при размяна на подводници, плуващи на позиции i и $i+1$, се променя само броят на сигналите, получавани от тях и, евентуално, на още една подводница, плуваща след тях в бойния ред.

Да обосновем това твърдение:

1. Очевидно е, че броят на сигналите, които би приела подводница, плуваща на позиция $k<i$ няма да се промени от това, че подводниците на позиции i и $i+1$ са разменили местата си – това следва от факта, че всяка подводница може да приема сигнали само от подводници, плуващи преди нея в бойния ред.
2. Нека с $d(k)$ означим дълбочината, на която плува подводницата, намираща се на позиция k, а с $s(k)$ – броя сигнали, които би получила при текущото разположение на подводниците. Лесно се съобразява, че сигнали може да приема само подводница, плуваща на такава позиция k, че плуващата непосредствено преди нея подводница е на по-малка дълбочина, т.е. $d(k)> d(k-1)$. Наистина, ако $d(k)< d(k-1)$, то подводницата на позиция $k-1$ не би пропуснала нито един сигнал към тази на позиция k.
3. Да разгледаме двата случая на размяна между позиции i и $i+1$:
   a. Преди размяната $d(i)>d(i+1)$



Преди размяната подводницата на позиция $i+1$ не е получавала никакви сигнали $(s(i+1)=0)$, а за тази на позиция i не се знае каква е била стойността на $s(i)$. След размяната, сигналите, които е получавала старата подводница на позиция i ще се поделят между двете подводници (подялбата може да е съвсем „по братски“ – едната нищо да не получи), а подводницата на позиция $i+1$, която плува по-дълбоко, със сигурност ще получава поне един сигнал – от подводницата на позиция i. С въпроса как да изчислим подялбата на сигналите ще се занимаем след малко, а сега да видим какво става със сигналите, получавани от подводниците, плуващи на позиции след $i+1$. Единствената промяна ще бъде, че, ако е имало подводница, получаваща сигнал от подводницата, плуваща на позиция $i+1$ преди размяната, то след размяната тя ще престане да го получава, т.е. броят на получаваните от нея сигнали ще намалее с 1. Така възниква първият въпрос, на който трябва да можем бързо да отговаряме:

Въпрос 1: При зададена позиция i, как бързо да се намира позицията j>i, такава, че подводницата, плуваща на позиция j, би получавала сигналите на подводницата, плуваща на позиция i? Всъщност въпросът е: при зададено i как бързо да се намира най-близкото j>i, такова че d(j)>d(i)?

Сега да се върнем на разпределението на сигналите между двете подводници след размяната. Нека $k<i$ е най-близката позиция преди позиция i, на която, след размяната, плува подводница на по-голяма дълбочина от подводницата на позиция i. Тогава подводницата на позиция i не би могла да получи сигнал от никоя подводница, плуваща на позиция $p \le k$. За броя на сигналите, получавани от подводницата на позиция i след размяната е валидна формулата:

$$s(i)=(i-k-1) - \sum s(q) \text{ по всички } k<q<i,$$

т.е. тя ще получава сигнали от всички подводници между позиция k и позиция i, чийто сигнали не се получават от някоя подводница от същото множество (да си спомним, че величините $s(q)$ при $q<i$ не са се променили при размяната). Броят на сигналите, получавани от подводницата на позиция $i+1$ след размяната, се дава от формулата:

$$s(i+1)=s'(i)-s(i)+1,$$

където $s'(i)$ е броят на сигналите, получавани от подводница на позиция i преди размяната.

Така възникват още два въпроса:

Въпрос 2:При зададена позиция i, как бързо да намираме най-близката позиция k, такава че k<i и d(k)>d(i)? Този въпрос е аналогичен на въпрос 1.

Въпрос 3:При зададени позиции k<i, как бързо да намираме сумата $\sum s(q)$ **по всички** $k<q<i$**?**

Отговаряйки на трите въпроса, ние ще можем да решим и подзадача 3 в този случай. Отговорите ще дадем след като разгледаме и втория случай на размяна, тъй като там възникват аналогични въпроси.

b. Преди размяната $d(i)<d(i+1)$

преди размяната *след размяната*

Този случай е по-лесен. Преди размяната всяка от подводниците е била в състояние да получава някакви сигнали – нека техният брой означим с $s'(i)$ и $s'(i+1)$. След размяната подводницата на позиция $i+1$ няма да получава никакви сигнали, т.е. $s(i+1)=0$. Подводницата на позиция i ще получава всички сигнали, които до размяната са получавали двете подводници без един – този който е получавала подводницата на позиция $i+1$ от подводницата на позиция i. С други думи $s(i)=s'(i)-s'(i+1)-1$. Остава да видим какво става с подводниците на позиции след $i+1$. Там подводницата, която, след размяната, е на най-близка позиция $j>i+1$, такава че $d(j)>d(i+1)$ (ако има такава) ще започне да получава още един сигнал от подводницата на позиция $i+1$. Тук сме в ситуацията, когато трябва да отговорим на Въпрос 1 от предния случай.

И така, ще решим задачата, като след всяка команда „размени позиции“ преизчисляваме броя сигнали, които би получавала всяка подводница при текущото разположение, а при получаване на команда „изпрати сигнал“ извличаме максималния брой за константно време. Преизчисляването ще се извършва със сложност $O((logN)^2)$.

За да отговаряме бързо на въпроси 1 и 2 ще поддържаме двоично индексно дърво, в листата на което стоят дълбочините на които плуват подводниците на всички позиции от 1 до N. Това индексно дърво ще се обновява след всяка команда „размени позиции“ и всеки възел, който не е листо, ще съдържа по-голямата от стойностите, записани в двамата му сина. Дървото ще се използва за изпълнение на заявки от типа Range Maximum Query, т.е. заявки „извлечи максималната дълбочина, на която плува подводница между позиции i и j“.

Тук е най-важният момент в решаването на задачата – изпълнението на RMQ заявките ще съчетаем с двоично търсене, за да намираме, за дадена позиция i, най-близката и отдясно или отляво позиция, на която плува подводница на по-голяма дълбочина от тази на позиция i, т.е. да отговаряме на въпроси 1 и 2.

Да разгледаме случая, когато търсенето е надясно от дадена позиция i. Първо пускаме една RMQ заявка от позиция $i+1$ до края, т.е. до позиция N. Ако върната дълбочина е по-малка от тази на подводницата на позиция i, то надясно няма по-дълбоко плуваща подводница. Ако е по-голяма, то има, и тогава делим интервала $[i+1,N]$ по средата и пускаме RMQ в левия подинтервал – ако там има по-голяма дълбочина, то продължаваме да делим него, иначе делим десния. Това продължава докато двете граници се срещнат в позицията, където плува на по-голяма дълбочина най-близката подводница. Важно е, че при разделяне на даден интервал на две, винаги пускаме RMQ първо в лявата му половина. Това е реализирано във функция ***find_right*** в програма ***submarines3***. Аналогично работи и функция ***find_left***, като в нея винаги пускаме RMQ първо в десния подинтервал. Обновяването на стойност в дадено листо на индексното дърво със съпътстващото обновяване до корена е със сложност $O(logN)$. Изпълнението на една RMQ заявка е също със сложност $O(logN)$. Като се има предвид, че прилагаме двоично търсене с изпълнение на RMQ, то отговорите на въпроси 1 и 2 се получават със сложност $O((logN)^2)$.

За намирането на сумите от броя на сигналите, които получават подводници на позиции от i до j, използваме друго индексно дърво, във всяко листо на което стои броят на сигналите, получавани от подводницата на съответната позиция. Това е класическа задача и тук няма да я разглеждаме подробно. Сложността на обновяване на едно листо и намиране на сума е $O(logN)$.

Конкретно в тази реализация, за намиране на максималния брой сигнали, които се получават от някоя подводница, използваме трето индексно дърво, във всяко листо на което отново стои броят на сигналите, получавани от подводницата на съответната позиция. Всеки баща съдържа по-голяма от двете стойности, записани в синовете му. В корена стои максималният брой сигнали. Обновяването на едно листо и съответните върхове до корена става със сложност $O(logN)$, а извличането на максимума е със сложност $O(1)$. Тук може да се използва и пирамида (heap).

Тъй като, както беше показано по-горе, при всяка команда „размени позиции" се сменя броя на получаваните сигнали най-много на 3 подводници, то обновяването на цялата информация след такава команда е със сложност $O((logN)^2)$. Получаването на максималния брой получавани от подводница сигнали след команда „изпрати сигнал" е със сложност $O(1)$. Цялото решение за изпълнението на M команди е със сложност $O(max(N,M*(logN)^2))$ и е дадено в програма **submarines3.**

Автори: Руско Шиков, Антон Шиков