

Un percorso alternativo, sebbene richieda 8 minuti e quindi non sia una soluzione valida, è il seguente:

- da $t = 0$ a $t = 1$: $\textcircled{0} \rightarrow \textcircled{1}$
- da $t = 1$ a $t = 2$: $\textcircled{1} \rightarrow \textcircled{4}$
- da $t = 2$ a $t = 3$: $\textcircled{4} \rightarrow \textcircled{5}$
- da $t = 3$ a $t = 5$: si rimane fermi in $\textcircled{5}$
- in $t = 5$: alta marea
- da $t = 5$ a $t = 6$: $\textcircled{5} \rightarrow \textcircled{7}$
- da $t = 6$ a $t = 7$: $\textcircled{7} \rightarrow \textcircled{6}$
- da $t = 7$ a $t = 8$: $\textcircled{6} \rightarrow \textcircled{8}$

È facile verificare che, nella mappa mostrata, non ci sono modi di arrivare in $N - 1$ in meno di 7 minuti.

Per risolvere il problema, dovete scrivere una funzione `solve(N, M, T, S, E)`, che restituisca un singolo numero intero: il minimo tempo necessario per arrivare dal punto 0 al punto $N - 1$, sapendo che l' i -esimo degli M canali collega i punti di intersezione S_i ed E_i , che la corrente inizialmente fluisce da S_i a E_i , e che l'alta marea arriva al minuto T . Se non è possibile andare dal punto 0 al punto $N - 1$, allora la funzione deve restituire il valore -1 .

Subtask

- **Subtask 0 [5 punti]:** caso di esempio.
- **Subtask 1 [11 punti]:** $2 \leq N \leq 100\,000$, $1 \leq M \leq 2\,000\,000$ e $T = 0$.
- **Subtask 2 [25 punti]:** $2 \leq N \leq 3\,000$, $1 \leq M \leq 20\,000$ e $0 \leq T \leq 1\,000\,000\,000$.
- **Subtask 3 [38 punti]:** $2 \leq N \leq 100\,000$, $1 \leq M \leq 2\,000\,000$ e $0 \leq T \leq 1\,000\,000\,000$.
- **Subtask 4 [21 punti]:** $2 \leq N \leq 200\,000$, $1 \leq M \leq 3\,000\,000$ e $0 \leq T \leq 1\,000\,000\,000$.

Dettagli di implementazione

Dovrai sottoporre esattamente un file con estensione: `.c`, `.cpp` o `.pas`. Questo file deve implementare la funzione `solve` utilizzando uno dei seguenti prototipi.

Programma in C o C++

```
int solve(int N, int M, int T, int* S, int* E);
```

Programma in Pascal

```
function solve(N, M, T: longint; var S, E: array of longint): longint;
```

La funzione implementata dovrà comportarsi come sopra descritto. Naturalmente sei libero di implementare altre routine ausiliarie per uso interno. La tua sottoposizione non deve effettuare direttamente operazioni di input o output, via console o via file di testo.

Funzionamento del grader di esempio

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati di input dal file `input.txt`, chiama la funzione che dovete implementare, e scrive il risultato restituito dalla vostra funzione sul file `output.txt`. Il file `input.txt` deve essere composto da $M + 1$ righe, in questo formato:

- nella prima riga devono essere presenti 3 interi: N , M , T ;
- nella i -esima delle successive M righe devono essere presenti 2 interi: S_i e E_i .

Esempio di input/output

File input.txt	File output.txt
9 10 5 0 1 1 4 4 2 2 0 2 3 4 5 7 5 6 7 6 3 8 6	7

Nota: l'esempio riportato è relativo alla situazione mostrata in figura.