

# ASD: Endgame (endgame)

## Testo del problema

Slides originali su: [judge.science.unitn.it/slides/asd20/prog2.pdf](http://judge.science.unitn.it/slides/asd20/prog2.pdf)

```
int main() { double result; ... out « scientific « setprecision(10) « E « ” “; out « scientific « setprecision(10) « G « ”
”; out « scientific « setprecision(10) « T « endl; ... } \end{lstlisting}
```

Saranno accettati gli output in cui le variabili  $E, G, T$  dichiarate nell'output si discostano meno di  $\frac{1}{10.000}$  dal valore calcolato usando  $p$  e  $t$  nell'Equazione ??;.

## Punteggio

- Ci sono 20 casi di test: ogni test assegna un punteggio di massimo 5 punti per un totale massimo teorico di 100 punti.
- Una soluzione è valida se rispetta tutte le richieste. Soluzioni non valide fanno **zero punti!**
- I percorsi validi ottengono questo punteggio:
- Il punteggio viene calcolato usando l'**ultima soluzione** terminata con tre asterischi \*\*\*.

I parametri di valutazione sono i seguenti:

- $G$  - energia contenuta nel guanto;
- $R$  - consumo di energia del guanto per unità di tempo;
- $T$  - tempo impiegato per visitare le città;
- $maxT$  - upper bound del tempo per visitare le città;
- $minT$  - lower bound del tempo per visitare le città;
- $maxG$  - energia ottima caricabile nel guanto;

Per ogni caso di test per cui la vostra soluzione fornisce un output entro i limiti di tempo e memoria otterrete il seguente punteggio  $P$ :

$$P = \frac{G + R(maxT - T)}{maxG + R(maxT - minT)} * 5 \quad (1)$$

**nota:** non è necessario che calcoliate il punteggio della vostra soluzione - vi servirebbero i parametri dei bound  $minT$ ,  $maxT$  e  $maxG$  che non avete. Il vostro obiettivo è in ogni caso di **massimizzare**  $E$ .

## Esempi (punteggio)

Nell'esempio di cui sopra, la risposta del sistema del valutazione è:

```
Soluzione valida: E(3.76) G(11.00) T(7.24) maxG(25.00) minT(1.09) maxT(35.00)
randMaxT(7.24) scoreRandMaxT(0.35) 0.353090
```

Il punteggio è calcolato nel modo seguente:

$$P = \frac{11 + 1 \cdot (7,2444444 - 7,2444444)}{25 + 1 \cdot (7,2444444 - 1,0909091)} \cdot 5 = 0,35308994 \cdot 5 \approx 1,77$$

## .1 Spiegazione del punteggio

Per ogni testcase, viene assegnato un punteggio da 0 a 1 *a* e poi moltiplicato per 5, secondo la formula di cui all'Equazione~1. Il messaggio resituito da arena indica i valori delle variabili usate nella formula. Questi valori vi permettono di valutare i punti di forza e i punti deboli della vostra soluzione.

Per esempio:

```
Soluzione valida: E(254800.15) G(365036.00) T(5511.79) maxG(366798.00) minT(3266.50)
                  maxT(729014.00) randMaxT(323342.21) scoreRandMaxT(0.99)
```

Le variabili stampate nel messaggio sono le seguenti:

- *E*: energia finale in output,  $G(p) - R \cdot T(p, t)$  (come in Eq. ??);
- *G*: energia contenuta nel guanto a fine percorso;
- *T*: tempo necessario per il percorso dichiarato;
- *maxG*: upper bound della quantità massima di energia che il guanto può contenere;
- *minT*: lower bound per il tempo necessario a compiere un giro;
- *maxT*: upper bound per il tempo necessario a compiere un giro;
- *randMaxT*: tempo richiesto da una soluzione che prende strade a caso, scegliendo pietre a caso;
- *scoreRandMaxT*: questo è il punteggio che la vostra soluzione ha ottenuto per il testcase corrente, che verrà poi moltiplicato per 5. In questo caso, verranno assegnati  $4.95$  ( $0.99 \times 5$ ) punti a questo test. Nel caso in cui il punteggio sia negativo, significa che il vostro percorso ha richiesto più tempo di *randMaxT*, e quindi il punteggio assegnato sarà 0.

## Valutazione

Per la valutazione del progetto:

- Conta il punteggio dell'**ultimo sorgente** inviato al sistema;
- Il progetto è superato con un punteggio non inferiore a 50 punti;
- C'è un limite di 40 sottoposizioni per gruppo;

## Limiti e assunzioni

- $1 \leq N \leq 2.000$ ,  $0 \leq S < N$
- $0 \leq M \leq 10.000$
- $0 \leq C \leq 10.000.000$
- $0 \leq R \leq 5.000$
- $0 \leq v_{\min} \leq v_{\max} \leq 1.000$
- $1 \leq m_i \leq 100.000$ ,  $1 \leq e_i < 100.000$
- Ogni grafo è completo.
- Ogni grafo è non diretto.

## Casi di test

- Ci sono 20 casi di test in totale.
- In almeno 3 casi su 20 in ogni città c'è una ed una sola pietra, non ci sono due città con pietre uguali.
- In almeno 3 casi su 20 tutti gli archi del grafo hanno lo stesso peso.
- In almeno 4 casi su 20 vale la disuguaglianza triangolare per i pesi degli archi del grafo.
- In almeno 10 casi su 20 non ci sono particolari limitazioni.

## Limiti delle risorse

- Tempo di esecuzione: 5 secondi (soft limit), 5,5 secondi (hard limit)
- Memoria: 64 MB

## Dataset di esempio

Per gli input forniti nel dataset di esempio non è stata calcolata una soluzione ottima. Per questo motivo il dataset non contiene anche i relativi output, solitamente messi a disposizione.

## Istruzioni di compilazione

Di seguito riportiamo le istruzioni per testare i vostri programmi su vari sistemi. Si suppone che il sorgente con il vostro codice si chiami file `endgame.cpp`. I file `endgame.cpp`, `grader.cpp` e `endgame.h` devono stare nella stessa cartella.

## Sistemi GNU/Linux

```
/usr/bin/g++ -DEVAL -std=c++11 -O2 -pipe -static -s -o endgame endgame.cpp grader.cpp
```

## Sistemi Mac OS X

Su sistemi Mac OS X usate il seguente comando di compilazione:

```
/usr/bin/g++ -DEVAL -std=c++11 -O2 -pipe -o endgame endgame.cpp grader.cpp
```

Se ottente un errore del tipo: `use of undeclared identifier quick_exit`, sostituite in `grader.cpp` l'istruzione `quick_exit(EXIT_SUCCESS);` con `exit(EXIT_SUCCESS);`.

## Sistemi Windows

Per il sistema Windows 10 potete installare il “Windows Subsystem for Linux”<sup>1</sup>. Successivamente potete installare i tool necessari per usare Visual Studio Code<sup>2</sup> o Visual Studio 2017<sup>3</sup> seguendo le relative guide riportate nelle note. Usando questo sistema fate attenzione a dove salvate i file e a quale nome gli date in quanto potreste avere delle difficoltà con percorsi che contengano spazi e caratteri speciali.

In alternativa, o per sistemi precedenti a Windows 10 potete installare *Cygwin*<sup>4</sup>, un ambiente completamente POSIX-compatibile per Windows. Anche in questo caso esistono guide per configurare i comuni editor disponibili su Windows di modo che utilizzino l'ambiente Cygwin, come per esempio Visual Studio<sup>5</sup>.

Una volta installato Cygwin è possibile simulare quanto avviene su arena compilando il proprio sorgente senza includere l'header `endgame.h` e il grader `grader.cpp`:

```
/usr/bin/g++ -DEVAL -std=c++11 -O2 -pipe -static -s -o endgame endgame.cpp
```

e lanciare il comando come:

```
timeout.exe 5 ./endgame
```

`timeout.exe` arresterà il programma dopo 5 secondi.

<sup>1</sup><https://docs.microsoft.com/en-us/windows/wsl/install-win10>

<sup>2</sup><https://code.visualstudio.com/docs/cpp/config-wsl>

<sup>3</sup><https://devblogs.microsoft.com/cppblog/targeting-windows-subsystem-for-linux-from-visual-studio/>

<sup>4</sup><https://www.cygwin.com/>

<sup>5</sup><https://devblogs.microsoft.com/cppblog/using-mingw-and-cygwin-with-visual-cpp-and-open-folder/>

## Esempi di input/output

File input.txt	File output.txt
<pre> 5 0 5 5 1.0 1.0 11.0 2 1 1 9 2 7 2 1 2 9 2 0 1 3 0 1 3 1 1 1 2 1 3 4 8 4 7 8 1 3 2 1 2 </pre>	<pre> 3.7555555556 11.0 7.2444444444 1 0 -1 2 -1 0 1 2 3 4 0 *** </pre>
File input.txt	File output.txt
<pre> 6 3 6 0 1 1.0 1.0 1 1 1 1 1 1 1 1 1 1 1 1 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 8 9 9 9 10 3 5 4 9 3 7 7 2 8 8 </pre>	<pre> -3.1000000000e+01 0.0000000000e+00 3.1000000000e+01 -1 -1 -1 -1 -1 -1 3 2 5 0 4 1 3 *** -2.7000000000e+01 0.0000000000e+00 2.7000000000e+01 -1 -1 -1 -1 -1 -1 3 2 5 0 1 4 3 *** </pre>

File input.txt	File output.txt
5 2 8 100 0 1.0 1.0 1 6 1 3 1 9 1 4 15 8 11 16 15 6 15 1 5 1 3 2 4 0 2 0 3 5 3 4 2 0 1 1 0 1 4 1 0 2 2 4 4 4 2 1 3 1 1 1 1 1 1 1 1 1 1	2.7000000000e+01 2.7000000000e+01 5.0000000000e+00 0 3 2 -1 4 -1 -1 1 2 0 1 3 4 2 *** 3.0000000000e+01 3.0000000000e+01 5.0000000000e+00 1 3 2 0 4 -1 -1 -1 2 1 3 0 4 2 *** 3.2000000000e+01 3.2000000000e+01 5.0000000000e+00 0 3 1 -1 4 -1 2 -1 2 0 1 3 4 2 *** 4.0000000000e+01 4.0000000000e+01 5.0000000000e+00 3 -1 2 -1 4 0 -1 1 2 0 3 1 4 2 *** 4.2000000000e+01 4.2000000000e+01 5.0000000000e+00 1 3 2 -1 4 0 -1 -1 2 1 0 3 4 2 *** 4.5000000000e+01 4.5000000000e+01 5.0000000000e+00 3 -1 1 -1 4 0 2 -1 2 1 0 3 4 2 ***