Prova Finale

Algoritmi e Strutture Dati

AA 2017/2018

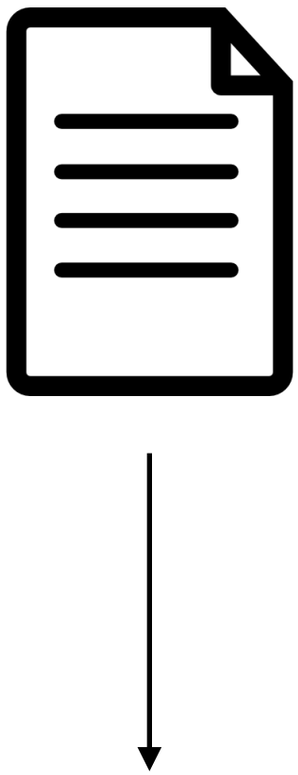
Struttura della prova

Implementazione in linguaggio C standard un interprete di Macchine di Turing non-deterministiche, nella variante a nastro singolo e solo accettori.

Un unico file sorgente contenente la vostra implementazione

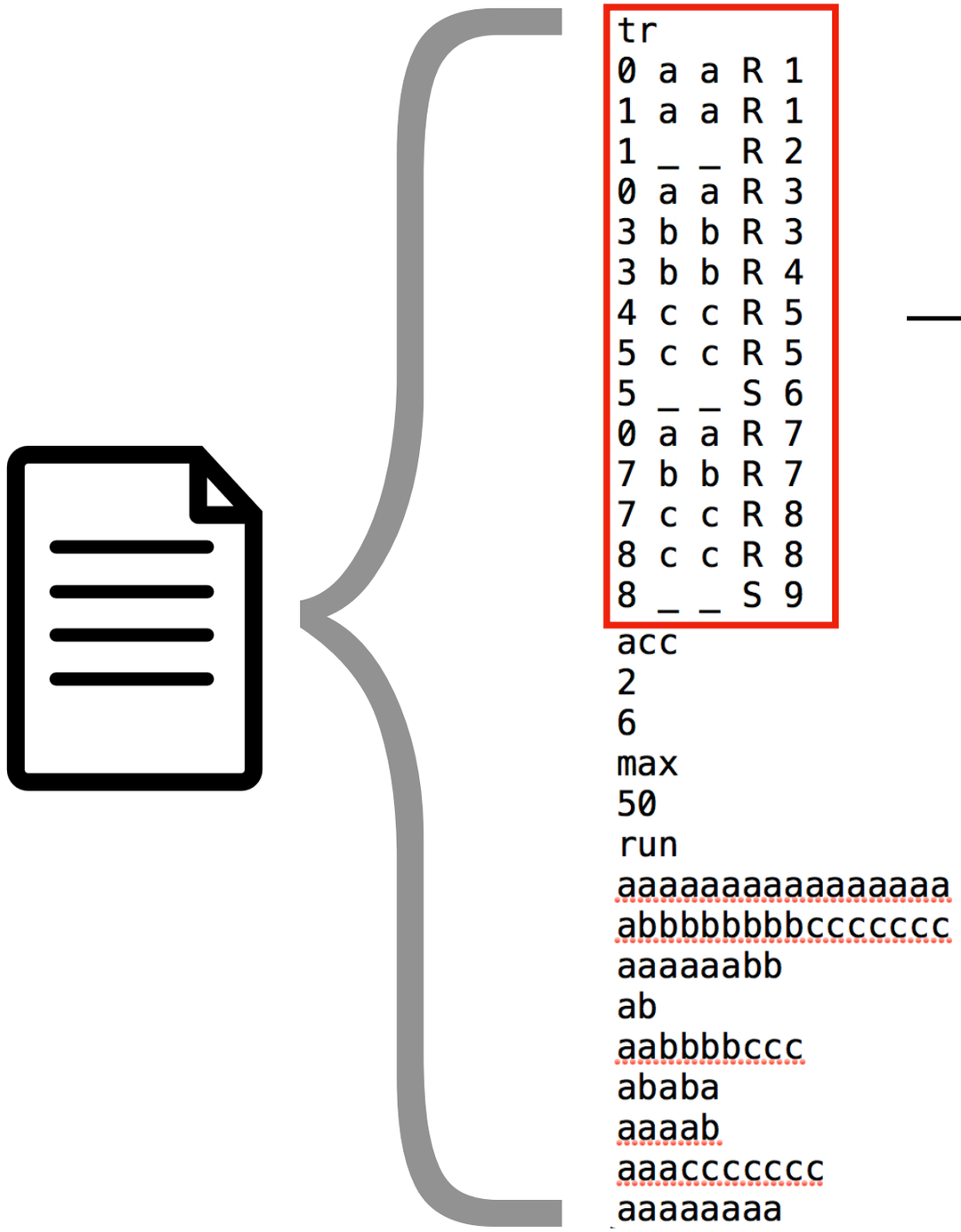
- NO librerie esterne eccetto standard (stdio, stdlib, string, math, ecc…)
- Ma posso usare questa libreria per… ? **NO**

Input



- Unico file di testo fornito tramite lo standard input
- Contiene:
  - funzione di transizione
  - stati accettazione
  - numero max di passi per una singola computazione
  - serie di stringhe da far leggere alla macchina

Input

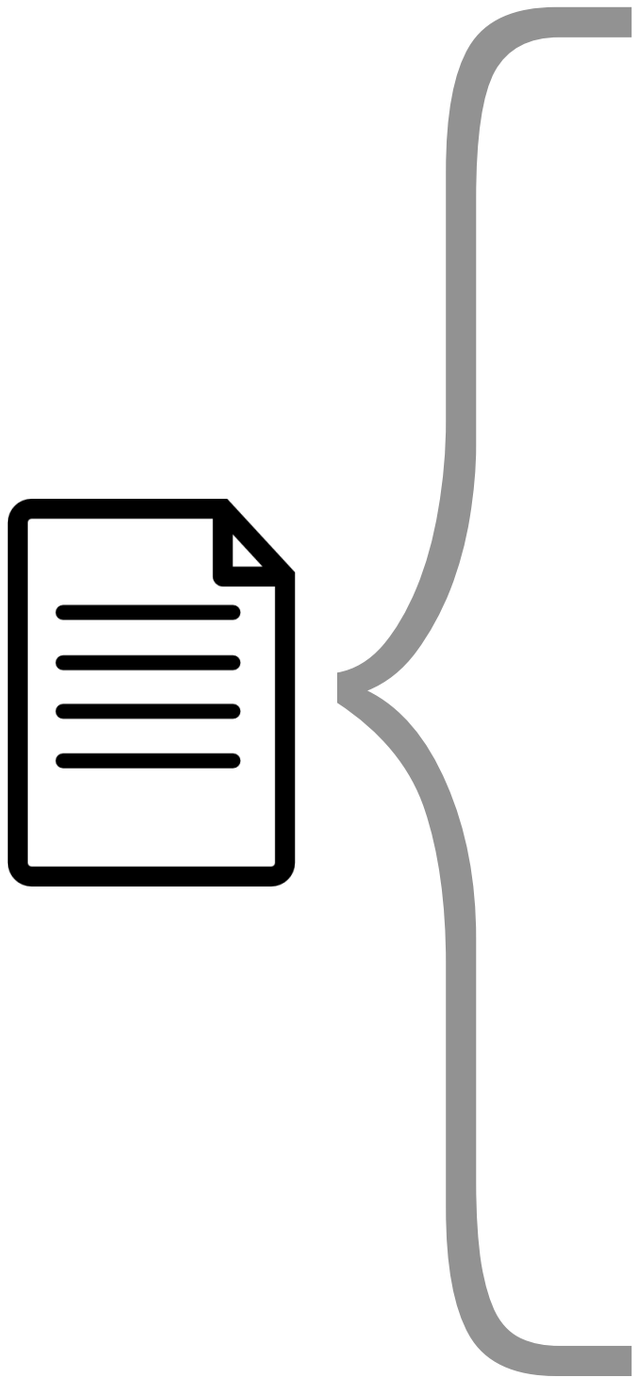


```
tr
0 a a R 1
1 a a R 1
1 _ _ R 2
0 a a R 3
3 b b R 3
3 b b R 4
4 c c R 5
5 c c R 5
5 _ _ S 6
0 a a R 7
7 b b R 7
7 c c R 8
8 c c R 8
8 _ _ S 9
acc
2
6
max
50
run
aaaaaaaaaaaaaaaa
abbbbbbbbccccccc
aaaaaabb
ab
aabbbbccc
ababa
aaaab
aaacccccccc
aaaaaaaa
```

Descrizione TM

- simboli di nastro sono dei *char*
- stati sono *int*
- "_" indica il simbolo "blank".
- I caratteri "L", "R", "S" indicano movimento della testina.
- La macchina parte sempre:
  - stato 0
  - primo carattere della stringa

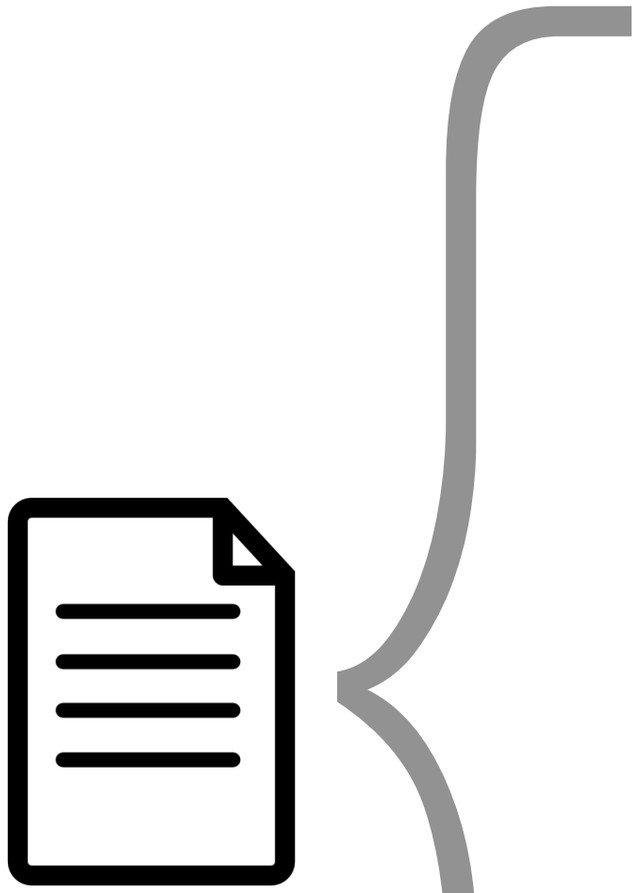
Input



```
tr
0 a a R 1
1 a a R 1
1 _ _ R 2
0 a a R 3
3 b b R 3
3 b b R 4
4 c c R 5
5 c c R 5
5 _ _ S 6
0 a a R 7
7 b b R 7
7 c c R 8
8 c c R 8
8 _ _ S 9
acc
2
6
max
50
run
aaaaaaaaaaaaaaaa
abbbbbbbbccccccc
aaaaaabb
ab
aabbbbccc
ababa
aaaab
aaacccccccc
aaaaaaaa
```

Stati Accettazione

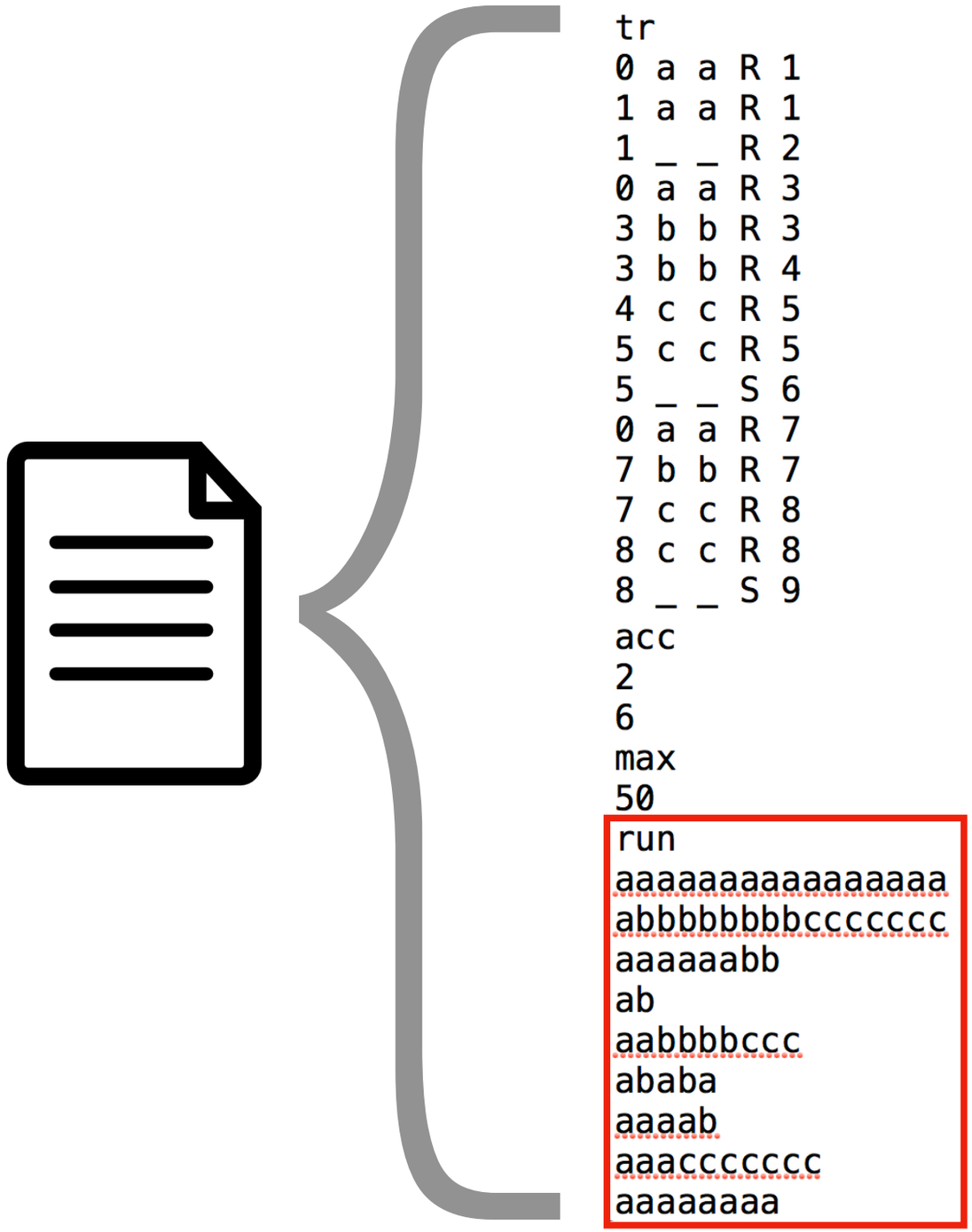
Input



```
tr
0 a a R 1
1 a a R 1
1 _ _ R 2
0 a a R 3
3 b b R 3
3 b b R 4
4 c c R 5
5 c c R 5
5 _ _ S 6
0 a a R 7
7 b b R 7
7 c c R 8
8 c c R 8
8 _ _ S 9
acc
2
6
max
50
run
aaaaaaaaaaaaaaaa
abbbbbbbbccccccc
aaaaaabb
ab
aabbbbccc
ababa
aaaab
aaacccccccc
aaaaaaaa
```

Numero Max Passi

Input

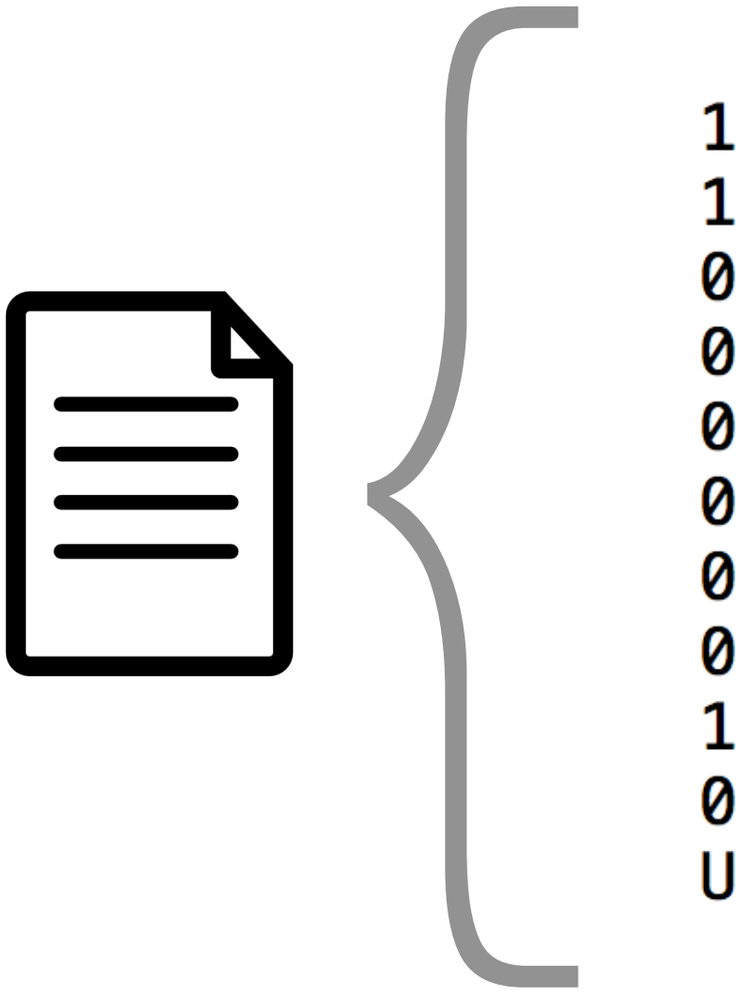


```
tr
0 a a R 1
1 a a R 1
1 _ _ R 2
0 a a R 3
3 b b R 3
3 b b R 4
4 c c R 5
5 c c R 5
5 _ _ S 6
0 a a R 7
7 b b R 7
7 c c R 8
8 c c R 8
8 _ _ S 9
acc
2
6
max
50
run
aaaaaaaaaaaaaaaa
abbbbbbbbccccccc
aaaaaabb
ab
aabbbbccc
ababa
aaaab
aaacccccccc
aaaaaaaa
```

Stringhe

Output

- Unico file di testo fornito tramite lo standard output



1 se stringa viene accettata
0 se stringa non viene accettata
U se non si e’ arrivati ad accettazione

Casi di Test Privati

- Test pubblici da intendersi di base
- Effettuati su centinaia di stringhe
- Stringhe di lunghezza arbitraria
- Verifica della correttezza dell’output
- Vincoli di tempo e memoria incrementali

Assumptions

- Potete assumere che i file di input siano sintatticamente corretti e coerenti con le specifiche
- La funzione di transizione può non essere ordinata per numero di stato
- Non ci saranno archi uscenti da uno stato di accettazione
- Se esiste lo stato N esistono anche gli stani N-1, N-2, …, 0
- Non ci sono vincoli riguardo alla lunghezza del file di input e delle stringhe di input
- Il parametro U in caso di macchina non-deterministica si riferisce al singolo percorso non-deterministico

Implementazione

Un unico file sorgente contenente la vostra implementazione

- Non iniziare subito a scrivere codice
- Iniziare ad impostare la soluzione (prima di settembre)
- Pensare alle strutture dati per soddisfare le specifiche sia funzionali che di complessità
- Il codice deve essere leggibile e ben commentato
- Sfruttare il paradigma procedurale (divide et impera)
- Esecuzione sequenziale (no multithreading)

Compilazione e Test in Locale

main.c

input_000.txt

output_000.txt

- Testare sempre in locale prima di caricare il codice (no brute force)

COMPILAZIONE

```
gcc -g -std=c11 -Wall -Werror -o main main.c
```

ESECUZIONE

```
cat input_000.txt | ./main > my_output.txt
```

CONTROLLO

```
diff output_000.txt my_output.txt
```