Prova Finale Algoritmi e Strutture Dati

Presentazione della consegna

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)
Politecnico di Milano

Obiettivi didattici e realizzazione

Obiettivi

- Applicazione pratica delle tecniche apprese nel modulo di algoritmi e strutture dati del corso di algoritmi e principi dell'informatica
- Implementazione di una soluzione ad un problema prestando attenzione ad aspetti concreti di efficienza del codice

Realizzazione

- Linguaggio C (C11, VLA ammessi)
- Nessuna libreria esterna al di là della libreria standard C
- No multithreading
- Dati in ingresso ricevuti via `stdin`, risultati da fornire via `stdout`

Modalità di realizzazione

- Il progetto è **strettamente individuale**
  - Non utilizzate **nulla generato da umano o macchina che non siate voi**
- Siete responsabili del vostro codice
  - Non caricatelo su repository pubblici
  - Non condividetelo con colleghi per "prendere ispirazione"
  - Non utilizzate alcun frammento di codice reperito su Internet
  - **Non utilizzate assistenti automatici come Claude, Copilot, ChatGPT e simili**
- In caso di plagi o uso di codice altrui, **tutti i progetti coinvolti** saranno annullati
- Per i casi dubbi o sospetti, sarà possibile un **orale integrativo**, in cui dovrete presentare le scelte progettuali effettuate.

Criteri di valutazione

Efficacia

- Correttezza ed efficienza della soluzione proposta sono valutate con batterie di test automatizzate
- Verranno forniti input/output d'esempio per poter collaudare la soluzione in locale
  - Non sottoponete soluzioni senza aver verificato che funzionino localmente
  - Verrà fornito anche uno strumento di generazione automatica di casi di test (input/output), per facilitarvi il testing in locale

Efficienza

- Il sistema di verifica calcola il tempo macchina e la memoria utilizzati
- La valutazione è immediatamente calcolata (e subito visibile), mediante 6 batterie di test (task, nel lessico del verificatore):
  - Ogni batteria ha una valutazione associata tra queste: $\{18, 21, 24, 27, 30, 30 \text{ e lode}\}$
  - Per ottenere una valutazione X è necessario e sufficiente superare la batteria di test con valutazione associata X

Criteri di valutazione - 2

- Nessun limite al numero di sottoposizioni, né penalità per sottoposizioni multiple
- È possibile migliorare la valutazione quante volte si desidera
- Avvertenza: viene valutata l'ultima sottoposizione fatta.
  - Sottoponete sempre il vostro sorgente definitivo al test in cui ottenete la valutazione più alta
- Verificatore disponibile all'indirizzo `https://dum-e.deib.polimi.it`
- Credenziali di accesso ricevute via mail istituzionale polimi
- Invio delle credenziali e apertura verificatore entro il 10 giugno

Scadenze e Pianificazione

- Per i laureandi di luglio:
  - 24 giugno 2026, ore 23.59 CEST. Segnalate (email al docente) la necessità di valutazione
- Per tutti gli altri:
  - 4 settembre 2026, ore 23.59 CEST, dopo di che la piattaforma verrà chiusa. Non segnalate la necessità di valutazione.
- Per laureandi di gennaio/febbraio (superati 145 CFU e iscritti all'esame di laurea):
  - la piattaforma sarà riaperta dal 29 gennaio al 12 febbraio.
- Iniziare a lavorare ad una settimana dalla scadenza è uno dei modi migliori per non riuscire a superare la prova

Tutoraggio

- Tutor suddivisi per fascia di cognome (ancora da determinare)
- Incontro con descrizione degli strumenti di sviluppo tenuto da A. Barenghi
  - 2026-06-12 dalle 14:15 alle 16:15 in aula 26.1.1.
  - Streaming: `https://politecnicomilano.webex.com/meet/alessandro.barenghi`
  - La registrazione sarà resa disponibile

Il Catering Impossibile

Devi organizzare una cena aziendale con tutti i dipendenti di una grande azienda. Ci sono diversi *piatti* possibili offerti dal ristorante identificati da un nome univoco.

- Ogni dipendente ha espresso delle richieste categoriche, dove ogni richiesta è una lista di opzioni, e al dipendente basta che almeno una delle sue opzioni venga soddisfatta (per esempio: *voglio la pizza oppure il sushi oppure che non ci sia il risotto*).
- Perché la serata sia un successo, tutti i dipendenti devono essere soddisfatti contemporaneamente. Il problema è capire *se esiste* un menu che non faccia lamentare nessuno.
- Il proprietario del ristorante si rende conto però che, dati i numeri sia delle richieste che dei dipendenti, potrebbe essere impossibile soddisfare tutti.
- Per questo motivo il ristoratore suggerisce all'organizzatore di mettere in ordine i dipendenti in base al livello gerarchico nella società: in caso di impossibilità, si procederà a considerare le richieste senza l'ultimo dipendente, poi scartando il penultimo e così via, fino ad arrivare ad una soluzione possibile.

Formato file in ingresso e stampe attese

- Il nome di un piatto è una stringa con le stesse convenzioni di formato di un nome di variabile `C` (e.g., `Risotto`, `pasta_al_pesto`).
- Ogni riga del file in ingresso indica la preferenza di un dipendente: il nome del piatto indica che il piatto è gradito; quando è preceduto dal carattere `-`, indica che il piatto è sgradito.
- Il file in ingresso inizia con una sequenza di nomi di piatti che indica i piatti disponibili; dalla riga successiva vengono elencate, una per riga, le richieste dei dipendenti, fornite come elenco di piatti, dal dipendente che occupa il posto più alto della gerarchia a quello che occupa il posto più basso.
- Per esempio la richiesta *voglio la pizza oppure il sushi oppure che non ci sia il risotto* corrisponde alla riga:
  `pizza sushi -risotto`.

Esempio di ingresso soddisfacibile

```
pizza sushi risotto torta
-pizza sushi
sushi risotto -pizza
```

In questo caso le richieste sono soddisfacibili, quindi in uscita si avrà:

```
OK
```

Esempio di ingresso insoddisfacibile

```
pizza sushi risotto torta
-pizza sushi -risotto
pizza -sushi risotto
-risotto
-sushi
risotto
sushi
```

In questo caso chiaramente le richieste non sono soddisfacibili, vista la discussione tra gli ultimi 4 dipendenti.

Esempio di uscita insoddisfacibile

L'uscita sarà:

```
KO
-1
-2
OK
```

`KO` perché non è in grado di soddisfare le richieste;
-1 significa che si prova senza la richiesta dell'ultimo dipendente;
-2 significa che non si è comunque riusciti, quindi si prova togliendo anche il penultimo;
a questo punto la soluzione è possibile: `OK`.