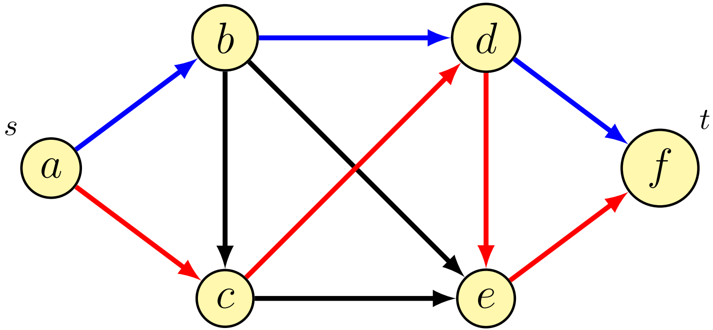
Cammini indipendenti (Esercizio 1.5 di 15-locale.pdf)

Dato un grafo orientato $G = (V, E)$ e due vertici $s, t \in V$, descrivere un algoritmo che restituisca la dimensione del più grande insieme di cammini edge-disjoint fra s e t.
Un insieme di cammini si dice edge-disjoint se ogni arco del grafo può comparire al massimo in uno dei cammini.



Cubi

Siano dati n cubi. Su ogni faccia del cubo è presente una lettera dell'alfabeto. Ogni cubo può essere descritto da una stringa di 6 caratteri (e.g. "ABCDEF"), che rappresenta le 6 facce del cubo. Sia data una parola costituita da $t \leq n$ caratteri. Descrivere un algoritmo che ritorni **true** se è possibile comporre la parola utilizzando t degli n cubi (scegliendo *una* faccia per ognuno di essi), **false** altrimenti. Discutere correttezza e complessità dell'algoritmo risultante.

Ad esempio, supponete di avere i seguenti cubi: "ABCDEF", "GHIJKL", "AABBCC", "ISTUVW" e di voler comporre la parola IDEA.

Torri di controllo (Esercizio 1.7 di 15-locale.pdf)

- Si consideri un insieme di aerei $A = \{a_1, \ldots, a_n\}$ e un insieme di torri di controllo $T = \{t_1, \ldots, t_m\}$.
- In un certo istante, ogni aereo a_i si trova alle coordinate $(a_i.x, a_i.y)$ e ogni torre t_j si trova alle coordinate $(t_j.x, t_j.y)$.
- Vincoli:
  - Carico: ogni torre può gestire al più L aerei;
  - Distanza: ogni torre può gestire aerei che si trovino al più a distanza d da essa.
- Descrivere un algoritmo che assegni ogni aereo ad una torre, rispettando i vincoli sul carico e sulla distanza.
- Discutere correttezza e complessità.

MC Donald

Gestire un McDonald non è semplice.

- La giornata lavorativa è suddivisa in 3 turni da 4 ore, dalle 11 alle 23.
- Durante il turno $t_i \in T$ (dove T l'insieme di 21 turni), è necessario che siano presenti p_i unità di personale
- Avete a disposizione un insieme D di dipendenti; ogni dipendente $d_j \in D$ dichiara un insieme di turni $n_j \subseteq T$ in cui non può lavorare.
- Ad esempio, d_j non può lavorare nei turni $\{t_1, t_7, t_{21}\}$.
- Per contratto aziendale, ogni lavoratore non può lavorare per più di 5 turni.
- Ogni giorno, un dipendendente non può lavorare per più di due turni (qualsiasi, anche non consecutivi).

Progettare un algoritmo che produca uno scheduling che illustri, per ogni turno, il personale associato e discuterne la complessità.

Spoiler alert!

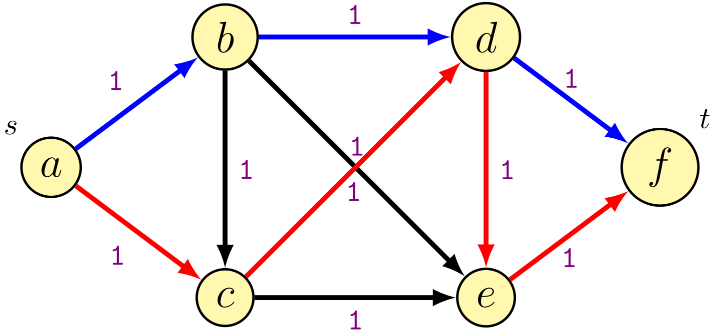
Cammini indipendenti (Esercizio 1.5 di 15-locale.pdf)

Si costruisce una funzione di capacità c tale per cui

$$c(x,y) = \begin{cases} 1 & (x,y) \in E \\ 0 & (x,y) \notin E \end{cases}$$

Si consideri il valore del flusso massimo flusso fra s (sorgente) e t (pozzo); questo valore corrisponde al numero totale di cammini indipendenti, in quanto essendo la capacità di tutti gli archi pari ad 1, ogni cammino aumentante avrà valore di flusso 1 e i suoi archi avranno capacità residua zero. In questo modo, ogni arco può essere utilizzato in un solo cammino.

Cammini indipendenti

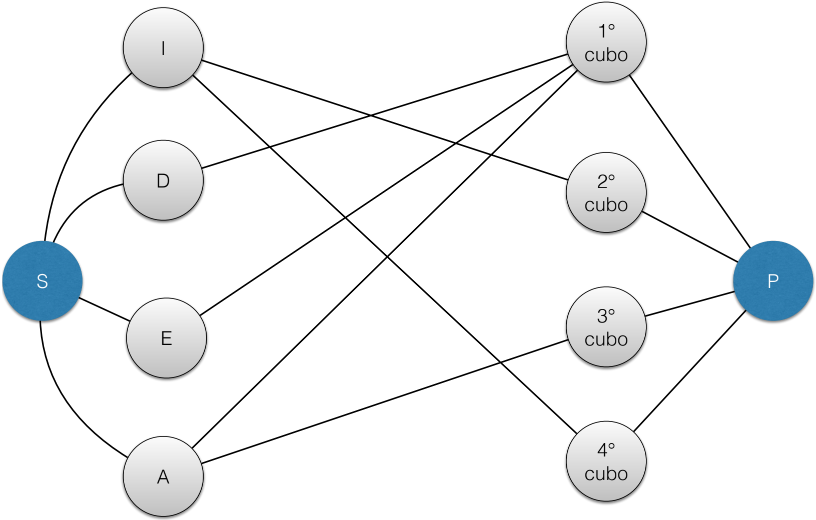


Cammini indipendenti

Complessità

- Seguendo il limite di Ford-Fulkerson, il costo dell'algoritmo è $O(|f^*|(n+m))$.
- $|f^*|$ è limitato da $O(n)$, perché ci sono al più $(n-1)$ archi che escono dalla sorgente (o che entrano nel pozzo)
- Quidi la complessità risultante è $O(n(m+n)) = O(mn)$.

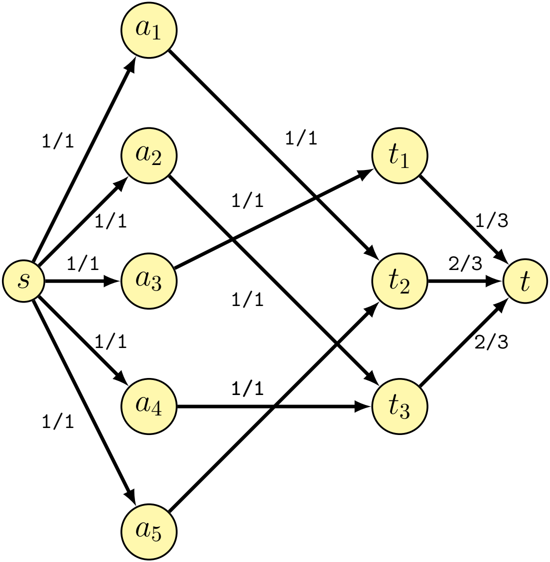
Cubi



Torri di controllo (Esercizio 1.7 di 15-locale.pdf)

- È possibile associare ad ogni aereo ed ad ogni torre i vertici di un grafo bipartito:
- L'insieme $V = \{s, t\} \cup \{a_1, \ldots, a_n\} \cup \{t_1, \ldots, t_m\}$ è costituito dai nodi sorgente, pozzo, i nodi aerei e i nodi torri.
- Nell'insieme E si inseriscono:
  - un arco (s, a_i) con capacità 1 fra la sorgente e ogni aereo a_i;
  - un arco (a_i, t_j) con capacità 1 se l'aereo a_i è entro la portata radio della torre t_j (ovvero, se $\sqrt{(a_i.x - t_j.x)^2 + (a_i.y - t_j.y)^2} \leq r$);
  - un arco (t_j, t) con capacità L fra ogni torre e il pozzo t.

Torri di controllo (Esercizio 1.7 di 15-locale.pdf)



Torri di controllo (Esercizio 1.7 di 15-locale.pdf)

- In questo modo:
  - ogni aereo viene assegnato al massimo a una torre di controllo;
  - ogni torre controllo può essere assegnato al massimo a L aerei.
- Gli archi il cui flusso è pari a 1 corrispondono agli assegnamenti aerei-torri. Il flusso massimo è limitato superiormente da n.
- Il numero di nodi è pari a $|V| = n + m + 2$;
- il numero di archi è pari a $|E| = n + m + O(nm)$.
- Utilizzando il limite di Ford e Fulkerson, il costo dell'algoritmo risultante è quindi $O(n(|V| + |E|)) = O(n^2m)$.