Grafi – Stessa distanza

- In un grafo orientato G, dati due nodi s e v, si dice che:
  - v è raggiungibile da s se esiste un cammino da s a v;
  - la distanza di v da s è la lunghezza del più breve cammino da s a v (misurato in numero di archi), oppure $+\infty$ se v non è raggiungibile da s
- Scrivere un algoritmo che prenda in input un grafo orientato $G = (V, E)$ e due nodi $s_1, s_2 \in V$ e che restituisca il numero di nodi in V tali che:
  - siano raggiungibili sia da s_1 che da s_2, e
  - si trovino alla stessa distanza da s_1 e da s_2.
- Discutere la complessità dell'algoritmo proposto.

Grafi – Grafi bipartiti

- Un grafo non orientato G è bipartito se l'insieme dei nodi può essere partizionato in due sottoinsiemi disgiunti tali che nessun arco del grafo connette due nodi appartenenti allo stesso sottoinsieme.
- $G = (V, E)$ è 2-colorabile se è possibile trovare una 2-colorazione di esso, ovvero un assegnamento $c[u] \in C$ per ogni nodo $u \in V$, dove C è un insieme di "colori" di dimensione 2, tale che:

  $$(u, v) \in E \Rightarrow c(u) \neq c(v)$$

- Si dimostri che G è bipartito:
  - se e solo se è 2-colorabile
  - se e solo se non contiene circuiti di lunghezza dispari
- Scrivere un algoritmo che prenda in input un grafo bipartito G e restituisca una 2-colorazione di G sull'insieme di colori $C = \{0, 1\}$, espressa come un vettore $c[1 \dots n]$. Discuterne la complessità.

Grafi – Distanza fra partizioni

- Dato un grafo G e due sottoinsiemi V_1 e V_2 dei suoi vertici, si definisce distanza tra V_1 e V_2 la distanza minima per andare da un nodo in V_1 ad un nodo in V_2, misurata in numero di archi.
- Nel caso V_1 e V_2 non siano disgiunti, allora la distanza è 0.
- Scrivere un algoritmo `mindist`(Graph G, Set V_1, Set V_2) che restituisce la distanza minima fra V_1 e V_2.
- Discutere complessità e correttezza, assumendo che l'implementazione degli insiemi sia tale che il costo di verificare l'appartenenza di un elemento all'insieme abbia costo $O(1)$.
- Nota: è facile scrivere un algoritmo $O(nm)$; esistono tuttavia algoritmi di complessità $O(n^2)$ (con matrice di adiacenza) e $O(m + n)$.

Grafi – Connetti il grafo

- Progettare un algoritmo efficiente che dato un grafo non orientato, restituisca il numero minimo di archi da aggiungere per renderlo connesso.
- Progettare un algoritmo efficiente che dato un grafo non orientato, aggiunga il numero minimo di archi necessari a renderlo connesso.