Esercizio 1 Si propone il seguente algoritmo per trovare, dato un grafo non diretto, connesso e pesato $G = (V, E, p)$ (p è la funzione peso), un minimo albero di copertura:

MST:
INPUT insieme di nodi $V = \{v_1, v_2, \ldots, v_n\}$, insieme di archi E, funzione peso p
  IF $|V| = 1$ THEN OUTPUT $\emptyset$
  ELSE
    $V_1 \leftarrow \{v_1, \ldots, v_{\lfloor n/2 \rfloor}\}$
    $V_2 \leftarrow \{v_{\lfloor n/2 \rfloor + 1}, \ldots, v_n\}$
    $E_1 \leftarrow \{\{v_i, v_j\} \in E \mid v_i, v_j \in V_1\}$
    $E_2 \leftarrow \{\{v_i, v_j\} \in E \mid v_i, v_j \in V_2\}$
    $CUT \leftarrow \{\{v_i, v_j\} \in E \mid v_i \in V_1 \text{ e } v_j \in V_2\}$
    Sia $\overline{e} \in CUT$ tale che $p(\overline{e}) = \min\{p(e) \mid e \in CUT\}$
    OUTPUT MST(V_1, E_1, p) $\cup$ MST(V_2, E_2, p) $\cup$ $\{\overline{e}\}$
  ENDIF
END.

1. *(max 5 punti)* Dimostrare che l'algoritmo MST restituisce sempre un albero di copertura del grafo di input.

2. *(max 12 punti)* Dire se l'algoritmo MST risolve il problema e in caso affermativo dimostrare la correttezza dell'algoritmo mentre in caso negativo fornire un controesempio.

Soluzione Esercizio 1

1. Possiamo provarlo per induzione sul numero n di nodi del grafo. Per $n = 1$ l'algortimo restituisce l'insieme vuoto e quindi l'asserto è vero. Assumiamo che, per grafi con al più n nodi, l'algoritmo restituisce un albero di copertura. Consideriamo ora un grafo $G = (V, E, p)$ di $n+1$ nodi, $G_1(V_1, E_1, p)$ e $G_2(V_2, E_2, p)$ sono grafi con al più n nodi quindi, per ipotesi induttiva l'insieme T_1 di archi restituito da MST(V_1, E_1, p) forma un albero di copertura per G_1 analogamente l'insieme T_2 di archi restituiti da MST(V_2, E_2, p) forma un albero di copertura per G_2. L'algoritmo per il grafo G restituirà l'insieme di archi $T = T_1 \cup T_2 \cup \{\overline{e}\}$ dove $\overline{e}$ è l'arco di peso minimo che connette nodi di V_1 a nodi di V_2 (tale arco esiste perchè il grafo G è connesso e quindi l'insieme CUT è non vuoto). Tra due qualsiasi degli $n+1$ nodi di G grazie agli archi in T c'è sempre un cammino infatti se i due nodi appartengono entrambi a V_1 o entrambi a V_2, l'esistenza del cammino è garantita dagli archi dell'albero di copertura T_1 o da quelli di T_2, rispettivamente, se invece i due nodi sono $a \in V_1$ e $b \in V_2$, il cammino che li congiunge parte da a, grazie agli archi dell'albero T_1 arriva ad un estremo dell'arco e, attraversa l'arco e e infine, grazie agli archi dell'albero T_2 raggiunge b. Quindi T è di copertura per G, resta da dimostrare che T non ha cicli, ma questo segue dal fatto che in T_1 ed in T_2 non ci sono cicli e l'introduzione del solo arco $\overline{e}$ non può produrne.

2. L'algoritmo non sempre produce un albero di copertura di peso minimo. Si consideri ad esempio il grafo completo di tre nodi dove

   $p(\{v_1, v_2\}) = 1$, $p(\{v_1, v_3\}) = 2$ e $p(\{v_2, v_3\}) = 10$.

   L'insieme dei vertici del grafo viene partizionato dall'algoritmo in $V_1 = \{v_1\}$ e $V_2 = \{v_2, v_3\}$ mentre per gli archi si ha $E_1 = \emptyset$, $E_2 = \{\{v_2, v_3\}\}$, $CUT = \{\{v_1, v_2\}, \{v_1, v_3\}\}$. Ora MST($V_1$, E_1, p) restituisce $\emptyset$, MST(V_2, E_2, p) restituisce $\{\{v_2, v_3\}\}$ e $\overline{e} = \{v_1, v_2\}$ L'algoritmo restituisce quindi la soluzione $\{\{v_1, v_2\}, \{v_2, v_3\}\}$ di peso 11 mentre la soluzione $\{\{v_1, v_2\}, \{v_1, v_3\}\}$ ha peso 3.

Esercizio 2 Il gestore di un cineclub propone ai soci l'insieme dei film che potrebbero essere proiettati e ciascun socio risponde indicando il sottoinsieme (non vuoto) dei film di suo gradimento. Siccome ogni film costa e il gestore non vuole scontentare nessuno, si trova ad affrontare il seguente problema: scegliere tra i film graditi ai soci un insieme di cardinalità minima che accontenta un po' tutti i soci, vale a dire, ogni socio gradisce almeno uno dei film scelti dal gestore. Viene proposto il seguente algoritmo:

INPUT sottoinsiemi dei film graditi ai soci $G_1, G_2, \ldots, G_n$ (il socio i gradisce i film in G_i)
    $SCELTI \leftarrow \emptyset$
    $F \leftarrow G_1 \cup G_2 \cup \cdots \cup G_n$
    $S \leftarrow \{1, 2, \ldots, n\}$
    WHILE $S \neq \emptyset$ DO
        $max \leftarrow 0$
        FOR ogni $f \in F$ DO
            $a \leftarrow |\{i \mid i \in S \text{ e } f \in G_i\}|$
            IF $a > max$ THEN
                $max \leftarrow a$
                $scelto \leftarrow f$
            ENDIF
        ENDFOR
        $SCELTI \leftarrow SCELTI \cup \{scelto\}$
        $S \leftarrow S - \{i \mid i \in S \text{ e } scelto \in G_i\}$
    ENDWHILE
OUTPUT $SCELTI$.

1. *(max 3 punti)* Dimostrare che l'algoritmo restituisce sempre un insieme di film che accontenta un po' tutti i soci.

2. *(max 10 punti)* Dire se l'algoritmo trova un insieme di cardinalità *minima* in grado di accontentare un po' tutti i soci e in caso affermativo dare una dimostrazione mentre in caso negativo fornire un controesempio.

Soluzione Esercizio 2

1. Notiamo che all'inizio l'insieme S contiene tutti i soci ed inoltre un socio viene eliminato da S solo dopo l'inserimento in $SCELTI$ di un film che lo soddisfa. Ne segue che qualora S risulti vuoto tutti i soci sono soddisfatti dai film inseriti in $SCELTI$. Resta solo da dimostrare che il ciclo WHILE termina.

   Ad ogni iterazione del WHILE in $SCELTI$ viene inserito il film segnalato dal maggior numero di soci in S e questi soci vengono poi eliminati da S. Ne consegue che ad ogni iterazione la cardinalità di S diminuisce e quindi dopo al più n iterazioni si avrà $S = \emptyset$ ed il ciclo termina.

2. L'algoritmo non sempre produce un insieme in grado di accontentare tutti i soci di cardinalità minima. Si considerino ad esempio le seguenti liste prodotte da 6 soci:

   $G_1 = \{a, b\}$, $G_2 = \{a, c\}$, $G_3 = \{a, d\}$, $G_4 = \{b\}$, $G_5 = \{c\}$ e $G_6 = \{d\}$.

   a è il primo film ad essere inserito in $SCELTI$ in quanto segnalato dal maggior numero di soci. In seguito per poter soddisfare i soci 4, 5 e 6 verranno inseriti in $SCELTI$ i film $\{b, c, d\}$. L'algoritmo produce quindi un insieme di cardinalità 4 mentre è facile vedere che l'insieme $\{b, c, d\}$ di soli 3 film soddisfa tutti i soci.