1. Pokażmy to zadanie nie wprost. Załóżmy, że w naszym grafie występują dwa drzewa MST T_1 i T_2.
   W jednym z drzew musi być krawędź o najniższej wadze, nazwijmy ją e i załóżmy, że znajduje się ona w T_1, ale nie w T_2. Niech ta krawędź znajduje się między wierzchołkami v_1 i v_2.
   Skoro krawędź e nie należy do T_2, to w T_2 musi istnieć inna ścieżka z v_i do v_j. Jeśli dołożymy teraz do T_2 krawędź e to powstanie nam tam cykl C. Podobna sytuacja wystąpi, jeśli dodamy pozostałe krawędzie cyklu C do drzewa T_1, dlatego w C musi istnieć jakaś krawędź, która jest w T_2, ale nie w T_1 – nazwijmy ją f.
   Skoro e ma najmniejszą wagę, to f musi mieć większą. Weźmy teraz nowe drzewo $T' = T_2 \cup \{e\}/\{f\}$. Wtedy to drzewo będzie miało mniejszą wagę od T_1 i T_2 i to ono będzie MST w G. Czyli otrzymaliśmy sprzeczność z założeniem.
2. Jeśli usunęlibyśmy jedną krawędź z dowolnego cyklu C w grafie G, to będzie on nadal spójny.
   Załóżmy, że cykl C składa się z krawędzi e_0, e_1, ..., e_{n-1}, gdzie:
   $c(e_0) > c(e_1) > ... > c(e_{n-1})$, gdzie $c(e_i)$ – waga krawędzi.
   Załóżmy, że wyrzucimy dowolną krawędź e_i z grafu, gdzie $i \in (0, n)$.
   Zauważmy, że wtedy dostaniemy drzewo T, które nie będzie MST grafu G, ponieważ jesteśmy w stanie zbudować inne drzewo T' usuwając z cyklu C krawędź najcięższą krawędź e_o, a wtedy $c(T) > c(T')$.
5. $c(e_i)$ – waga krawędzi e_i
   Do wyznaczenia drzewa rozpinającego używamy algorytmu Boruvki. Załóżmy nie wprost, że w jakiejś iteracji powstanie nam cykl.
   Weźmy wszystkie wierzchołki, które tworzą ten cykl i nazwijmy je kolejno: v_o, v_1, ..., v_{n-1}, gdzie n to długość cyklu.
   Kolejne wierzchołki tworzą między sobą krawędź, czyli v_i tworzy krawędź z v_{i+1} itd. Skoro jest to cykl to v_{n-1} utworzy krawędź z v_0. Oznaczmy te krawędzie jako: e_0, e_1, ..., e_{n-1}, gdzie e_i oznacza krawędź z v_i do v_{i+1}, a e_{n-1} krawędź z v_{n-1} do v_0.
   Wiemy, że wagi krawędzi w grafie są różne, więc z założenia algorytmu Boruvki otrzymujemy:
   $c(e_o) > c(e_1) > ... > c(e_{n-1}) > c(e_o)$, z czego wynika, że $c(e_o) > c(e_o)$. Sprzeczność, więc w żadnej iteracji algorytmu nie powstanie nam cykl.
6. e_i – dowolna krawędź,
   $c(e_i)$ – waga krawędzi e_i.
   Algorytm Boruvki działa tak, że dla każdego wierzchołka wybieramy tę krawędź e_i (wszystkie wagi są różne), że $c(e_i)$ jest najmniejsze. Zmodyfikujmy ten algorytm tak, aby działał w sytuacji, gdy jakieś krawędzie mają takie same wagi. Należy wówczas wprowadzić dodatkowo wagę $g(e_i)$ taką, że:

dla dowolnych krawędzi e_i, e_j:

- $c(e_i) < c(e_j)$: $g(e_i) < g(e_j)$.
- $c(e_o) = c(e_j)$, $i < j$: $g(e_i) < g(e_j)$.

Wtedy algorytm wybierałby taką krawędź e_i, że $g(e_i)$ jest najmniejsze. Tzn. bierzemy zbiór wszystkich krawędzi i sortujemy je po ich wadze, następnie indeksujemy je. Jeśli podczas wybierania krawędzi dla dowolnego wierzchołka kilka krawędzi będzie miało tę samą wagę, to wybierzemy tę z mniejszym indeksem. Dowód analogiczny do dowodu z zadania 5.

Załóżmy, że w grafie pojawił się cykl. Otrzymujemy:

$g(e_o) > g(e_1) > ... > g(e_{n-1}) > g(e_o)$, czyli sprzeczność.

10.

	Ania	Bartek	Cezary	Dąbrówka	Elwira
Skrzypce	X				X
Harfa	X	X		X	X
Kontrabas	X				X
Wiolonczela	X				X
Fortepian		X	X		

Mamy do obsadzenia 5 instrumentów oraz 5 osób. Cezary i Dąbrówka potrafią grać tylko na jednym z instrumentów, więc Cezary musi grać na fortepianie, a Dąbrówka na harfie. Skoro oba te instrumenty są już obsadzone to Bartek nie ma już na czym grać. Pozostały trzy instrumenty i dwie osoby.

Mamy zbiory: A – zbiór osób, B – zbiór instrumentów.

B' = {skrzypce, kontrabas, wiolonczela},

A' = {Ania, Elwira}.

$|A'| < |B'|$, więc warunek Halla nie jest spełniony, czyli tym osobom nie uda się utworzyć składu.