

# Zadanie 10 - Górską krainą zimą w czasach niedoli (czyli Backtracking)

---

author: PW

## Opowieść o górskiej krainie

W pewnej górskiej krainie w czasach dobrobytu osiedliło się dużo ludzi. Wybudowali oni sobie domy i żyli tworząc małe miasteczko.

Gdy dla owej krainy zaczęły się trudne czasy, część ludzi wyjechała i pozostawiła swoje domy opuszczone.

W czasie zimy grupa odpowiedzialna za odśnieżanie dróg dojazdowych postanowiła nie odśnieżać wszystkich dróg, a jedynie te które prowadzą do domów które wciąż są zamieszkane - w taki sposób aby były połączone ze sobą wszystkie jeszcze zamieszkane domostwa. Zadanie to można zrealizować na wiele sposobów, ale interesuje nas ten najbardziej oszczędny - wymagający odśnieżenia dróg o najmniejszej możliwej sumarycznej długości.

## Reprezentacja

Graf  $g$  reprezentuje ową krainę. Graf jest nieskierowany. Wierzchołki są domostwami, a krawędzie są drogami łączącymi domostwa. Długość drogi reprezentowana jest w grafie wagą krawędzi. Tablica `terminalVertices` oznacza domostwa które są w danym czasie wciąż zamieszkane.

Zadaniem jest napisanie algorytmu, który wyznaczy minimalną możliwą sumę długości dróg, które należy odśnieżyć, aby wszystkie zamieszkane domostwa były ze sobą połączone.

Dojazd do pozostałych domostw może być odśnieżony ale nie musi.

## Uwagi

- W ramach zadania trzeba napisać jeden wariant metody do wszystkich etapów. Kolejne etapy nie mogą psuć wyników poprzednich.
- Zadanie należy rozwiązać stosując algorytm z nawrotami (backtracking); inne rozwiązania nie będą uznane.
- Algorytm powinien działać w czasie  $O^*(2^m)$  i pamięci  $O(m)$ , gdzie  $m$  to liczność zbioru krawędzi.
- Każdą drogą wystarczy odśnieżyć raz; jeśli odśnieżarka będzie musiała przejechać przez jakąś drogę dwa razy, tylko za pierwszym razem uwzględniamy jej długość w koszcie rozwiązania.
- Przejście testów wydajnościowych w części domowej może wymagać rozpoczęcia obliczeń od znalezienia rozwiązania przybliżonego, a następnie odrzucania wszystkich częściowych rozwiązań o wyższym koszcie

## Punktacja

1. [1.5 p.] Zwracanie minimalnej długości dróg, dzięki której można połączyć `terminalVertices`

2. [1 p.] Zwracanie w wyniku zbioru użytych dróg (krawędzi).

Powinny być one zwracane w postaci tablicy użytych krawędzi `Edge[]`

## Przykłady

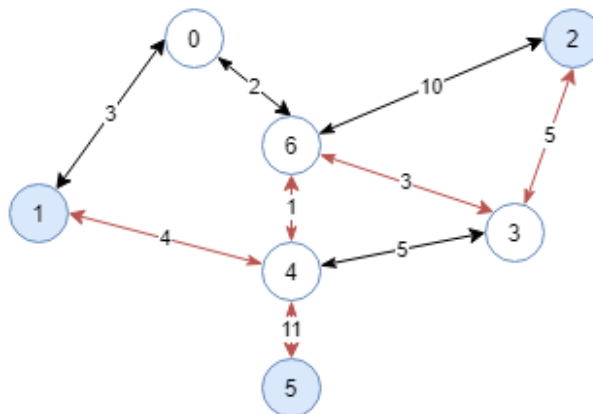
Poniżej znajdują się lista krain użyta w etapach 1 i 2.

Czerwona ścieżka oznacza **minimalną drogę** (bądź jedną z minimalnych, gdyż niektóre przypadki mogą mieć kilka równoważnych rozwiązań).

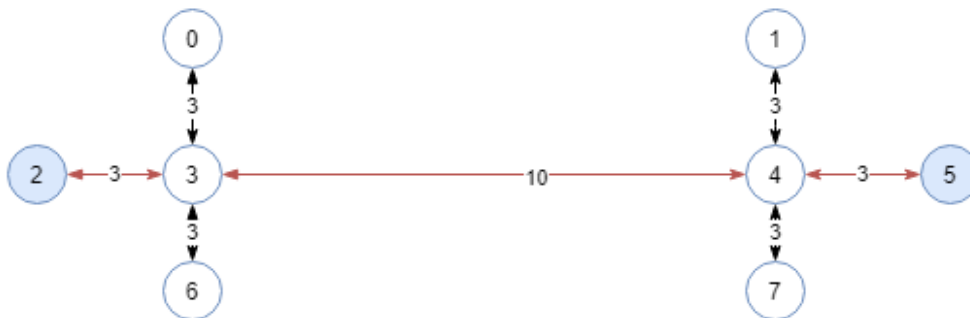
Simple



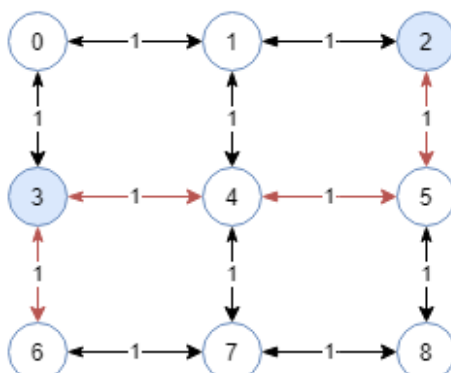
Small irregular graph



Barbell



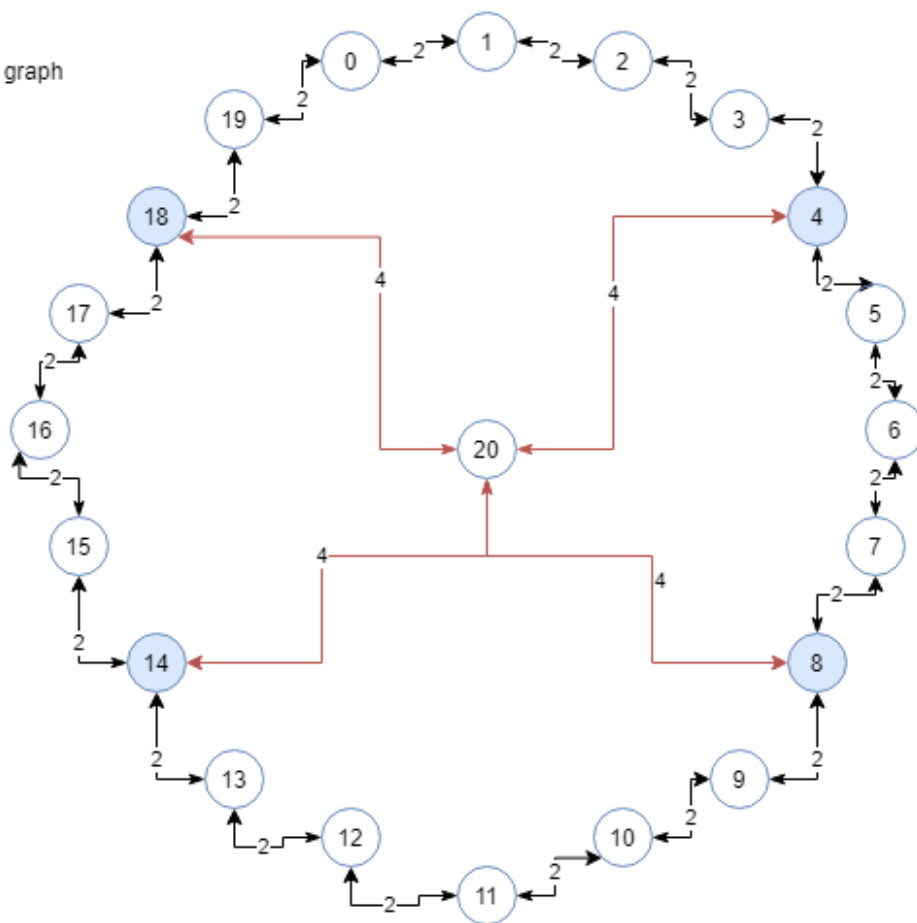
3x3 dense graph, with weights equal to 1



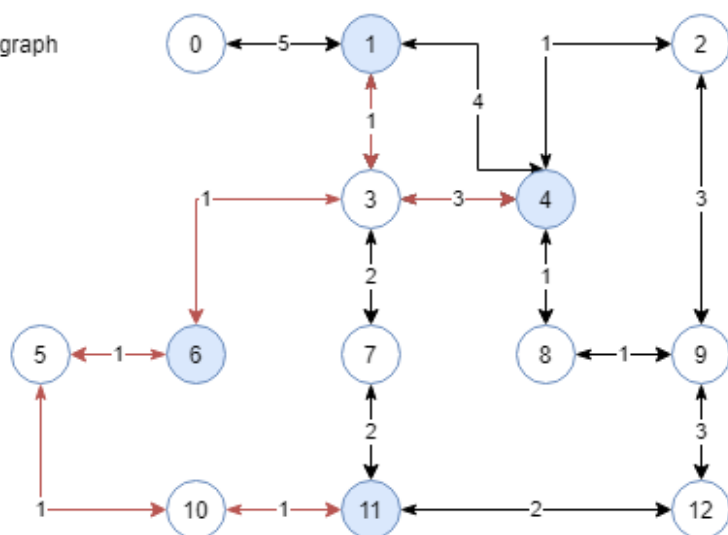
Graph with isolated vertices



Circle graph

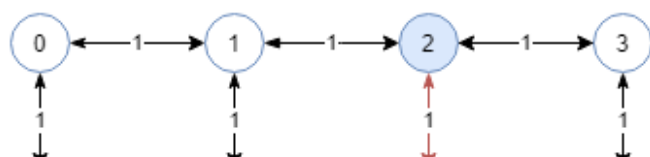


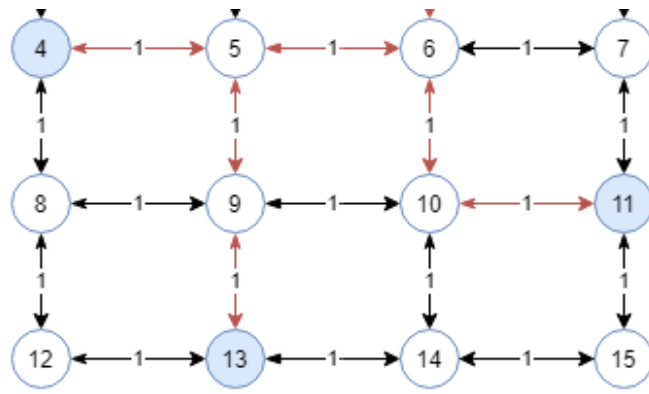
Irregular graph



## LARGE TESTS

4x4 dense graph,  
with weights equal  
to 1





3x10 dense graph,  
with weights equal  
to 1 or 2

