

# PIZZO

Vertex-Cover to problem, dla danego grafu i liczby  $k$ , czy można wybrać  $k$  wierzchołków tego grafu tak, by każda z krawędzi miała wybrany co najmniej jeden z końców. Zbiór wierzchołków, które pokrywają przynajmniej jeden z końców każdej krawędzi, będziemy nazywać *pokryciem*.

## Zadanie domowe nr 5 (6 punktów)

To zadanie można robić w parach lub samemu. W przypadku samodzielnego wykonania, należy pominąć części E i G.

**Część A.** Stwórz program, który dla zadanej liczby  $n$  tworzy losowy graf o  $n$  wierzchołkach. Zadbaj o to, żeby generowane grafy były zróżnicowane.

**Część B.** Stwórz program, który dla danego grafu robi *szybki szacunek* minimalnego pokrycia. Ma to być algorytm zachłanny, który wybiera wierzchołek o największym stopniu, dodaje go do prowizorycznego pokrycia i usuwa go (wraz z jego krawędziami) z grafu, powtarzając to aż do momentu, gdy otrzymamy pokrycie.

**Część C.** Wykorzystaj program z części A do wygenerowania grafów do testów. Dla tych grafów, narzędziem z części B oszacuj minimalne pokrycie, a następnie z prawdopodobieństwem 0.5 **od wyniku odejmij 1** (szacunek mógł być zbyt duży, a nawet jak nie był, to potrzebne nam do testów też instancje negatywne). W ten sposób przygotuj dane do testów programów rozwiązujących problem vertex cover. Upewnij się, że w danych są też duże wartości liczby  $k$  – jeśli nie ma, to popracuj nad zróżnicowaniem w części A.

**Część D.** Stwórz program, który rozwiązuje problem vertex-cover metodą brute-force, sprawdzając wszystkie podzbiory wierzchołków rozmiaru  $k$ .

**Część E.** Stwórz program, który rozwiązuje problem vertex-cover w czasie  $O(2^k|G|)^1$ .

**Część F.** Stwórz program, który rozwiązuje problem vertex-cover z wykorzystaniem smt-solvera.

**Część G.** Stwórz program, który rozwiązuje problem vertex-cover z wykorzystaniem sat-solvera.

**Część H.** Przeprowadź miarodajne eksperymenty porównujące czasy działania wszystkich stworzonych programów na grafach wygenerowanych w części C. Jak się zmieniają te czasy w zależności od rozmiaru grafu, a jak w zależności od liczby  $k$ ?

**Część I.** Dla każdego programu eksperymentalnie określ, jak duże grafy i liczby  $k$  jest w stanie przetworzyć w dwie minuty.

**Część R.** Przygotuj pojedynczy plik PDF z raportem z wykonania tego zadania. W nim opisz szczegółowo swoje rozwiązanie części A, F, G, H i I. Oprócz tabel i wykresów, należy wyciągnąć wnioski z eksperymentów.

Jako rozwiązanie wyślij w skosie dwa pliki: PDF (nie zapomnij podać autora/autorów w nagłówku) oraz archiwum z wszystkimi implementacjami.

---

<sup>1</sup>Wskazówka: Dla każdej krawędzi, w pokryciu musi być jej przynajmniej jeden koniec. Zauważ, że ten algorytm dla ustalonego  $k$  jest liniowy.