

WSI - ćwiczenie 1: Przeszukiwanie przestrzeni

3 marca 2026

1 Sprawy organizacyjne

1. Ćwiczenie realizowane jest samodzielnie w języku Python.
2. Ćwiczenie powinno zostać oddane najpóźniej do 17.03.2026r. W ramach oddawania ćwiczenia należy zademonstrować prowadzącemu działanie kodu i wnioski z raportu.
3. Przed oddaniem, kod oraz raport powinien być spakowany i wysłany do prowadzącego prowadzącego:
 - **grupa 102:** pawel.zawistowski@pw.edu.pl,
 - **grupa 106:** tomasz.radzikowski@pw.edu.pl.
4. Raport powinien być w postaci pliku `.pdf`, `.html` albo być częścią notebooka jupyterowego. Powinien zawierać opis eksperymentów, uzyskane wyniki wraz z komentarzem oraz wnioski.
5. Na ocenę wpływa poprawność oraz jakość kodu i raportu.
6. Można korzystać z pakietów do obliczeń numerycznych, takich jak `numpy`.
7. Implementacja powinna być ogólna – w szczególności, należy oddzielić logikę algorytmu przeszukiwania od specyfiki problemu, tak aby można było łatwo zastosować ten sam algorytm do innego problemu.

2 Ćwiczenie

Rozważamy problem robienia zakupów z ograniczonym budżetem B , w którym dane jest n produktów. Dla każdego produktu i znamy cenę $c_i > 0$ oraz wartość rangi $r_i > 0$, określającą jak bardzo dany produkt jest potrzebny. Każdy produkt może zostać kupiony co najwyżej raz.

Celem jest wybranie podzbioru produktów tak, aby łączna cena nie przekroczyła budżetu B , a suma rang była możliwie największa. Należy przy tym:

- zamodelować przestrzeń stanów jako drzewo decyzji (biore / nie biore kolejne produkty),
- zaimplementować dwa algorytmy: *najpierw najlepszy* (best-first) i A^* do przeszukiwania tej przestrzeni stanów,
- porównać skuteczność algorytmów pod względem jakości rozwiązania (uzyskana suma potrzebności) oraz liczby rozwiniętych stanów, dla dostarczonego pliku z danymi wejściowymi w formacie JSON.

Dane wejściowe (`shopping.json`):

```
{
  "budget": 1000,
  "item_costs": [
    220,208,198,192,180,180,165,162,160,158,155,130,125,122,120,118,115,110,105,101,100,
    100,98,96,95,90,88,82,80,77,75,73,72,70,69,66,65,63,60,58,56,50,30,20,15,10,8,5,3,1
  ],
  "item_ranks": [
    80,82,85,70,72,70,66,50,55,25,50,55,40,48,50,32,22,60,30,32,40,38,35,
    32,25,28,30,22,25,30,45,30,60,50,20,65,20,25,30,10,20,25,15,10,10,10,4,4,2,1
  ]
}
```