Problém batohu řešený hrubou silou a jednoduchou heuristikou

PAA – Úloha 1

# Specifikace úlohy

## Zadání

Naprogramujte řešení problému batohu hrubou silou (tj. exaktně). Na zkušebních datech pozorujte závislost výpočetního času na *n*.

Naprogramujte řešení problému batohu heuristikou podle poměru cena/váha. Pozorujte závislost výpočetního času na *n* a průměrnou a maximální relativní chybu (tj. zhoršení proti exaktní metodě).

## Problém batohu obecně

Jedná se o optimalizační problém, kde musíme splnit a zároveň maximalizovat kde , je binární vektor příslušnosti prvků v řešení, je vektor obsahující váhy prvků 1…n a vektor vah prvků. je maximální kapacita batohu.

## Rozbor možných variant řešení

## Hrubá síla

Hrubou silou procházíme všechna možná řešení problému a pouze vybereme to nejlepší z nich. Jelikož je prvků *n*, máme 2^n možných řešení.

Variantou řešení je tyto kombinace reprezentovat jako n-bitový čítač a pro každé jeho ohodnocení zkontrolovat nejlepší cenu, kde každý *m*-tý bit čítače reprezentuje příslušnost *m*-tého prvku ve stávajícím řešení.

Druhou variantou je reprezentovat stavový prostor jako binární strom, kde levé větve reprezentují přidání prvku do stávajícího řešení a pravé jeho nepřidání. Tento strom lze procházet do hloubky, rekurzivně nebo pomocí fronty, nebo do šířky.

## Heuristika

Jednoduchá heuristika upřednostňuje prvky s lepším poměrem cena/váha.

Toto lze zajistit seřazením všech prvků podle kritéria řadící funkce poměr cena/váha a potom ze seřazeného pole prvky postupně přidávat do řešení od jeho začátku.

Druhou variantou je všechny prvky přidat do prioritní fronty, kde priorita je opět nastavena jako poměr cena/váha. Prvky přidáváme do řešení odebíráním ze začátku fronty.

# Rámcový popis postupu řešení

## Hrubá síla

Algoritmus rekurzivně prohledává kompletně celý stavový prostor do hloubky a zaznamenává nejlepší řešení. Stavovým prostorem je binární strom, kde levé větve reprezentují přidání prvku do stávajícího řešení a pravé jeho nepřidání.

Jelikož máme *n* prvků, a každý z nich v řešení je nebo není, konfigurační proměnnou je binární vektor velikosti počtu prvků, značící jejich příslušnost(1), resp. absenci (0) v řešení. Složitost algoritmu je tedy O(2n).

## Heuristika

Jednoduchá heuristika bere v potaz poměr cena/váha každého prvku. Algoritmus tedy prvky podle tohoto poměru seřadí a pak postupně přidává prvky do řešení postupně od nejlepšího, dokud ještě lze do batohu nějaký prvek přidat. Algoritmus nezaručuje nalezení optimální ceny.

Existuje n prvků, všechny musíme seřadit podle jejich poměru cena/váha a poté lineárně projít toto seřazené pole a přidat prvky do řešení. Složitost je tedy O(n\*log(n) + n), čili O(n\*log(n)).

# Popis algoritmu

## Hrubá síla

Rekurzivní funkce dostává jako parametry současnou váhu a současnou cenu momentálně zkoumaného řešení (uzlu stromu) a pořadí momentálně zkoumaného prvku (úroveň ve stromu). Funkce zavolá sama sebe s upravenými parametry – přidání váhy i ceny současného prvku ke stávajícímu řešení a posunutí pořadí(přesunutí do levého podstromu) a zachování současné ceny i váhy a posunutí pořadí(přesunutí do pravého podstromu). Funkce se ukončí, pokud se nachází v listu stromu - zároveň zkontroluje, jestli je momentální řešení to nejlepší.

Stručně naznačená hlavní kostra rekurzivního algoritmu, pomoci pseudokódu:

Řeš rekurzivně(současná cena, současná váha, pořadí prvku){

Pokud v listu stromu(pořadí == počet prvků), porovnej výsledek s dosavadním nejlepším, ukonči.

Řeš rekurzivně(současná cena + cena prvku, současná váha + váha prvku, pořadí prvku + 1)

Řeš rekurzivně(současná cena, současná váha, pořadí prvku + 1)

}

## Heuristika

Heuristická funkce nejdříve seřadí prvky vybraným řadícím algoritmem, a poté v cyklu prochází od prvního do posledního prvku. Pokud přidání právě procházeného prvku do řešení nepřekročí maximální povolenou váhu batohu, prvek se přidá.

Stručné naznačená hlavní kostra heuristického algoritmu, pomocí pseudokódu:

Seřaď prvky()

Současná váha = 0

Opakuj pro všechny prvky

Pokud současná váha + váha prvku nepřevyšuje maximální nosnost batohu,

přidej prvek do řešení, přičti jeho váhu k současné váze

# Naměřené výsledky

# Závěr