# Algorithmen und Datenstrukturen II
# SoSe 2024 – Serie 10

## 1 0/1-Hamsterproblem

Ein Hamster[1] möchte die Kapazität seines Maules von $c = 900$[mg] möglichst gut ausnutzen, um für schlechte Zeiten vorzusorgen. Er hat gemäß folgender Tabelle Leckereien zur Auswahl, die er bereits mit indiviudellen Leckerheitsgraden versehen hat. Jede Leckerei kann nur gar nicht (0) oder vollständig (1) 'gehamstert' werden.

| Objekt $i$ | Gewicht $t_i$ [mg] | Leckerheitsgrad $p_i$ |
|---|---:|---:|
| 1: Sellerie | 600 | 10 |
| 2: Sonnenblumenkerne | 100 | 15 |
| 3: Rote Beete | 800 | 20 |
| 4: Hirse | 100 | 5 |
| 5: Buchweizen | 200 | 15 |

Benutzen Sie Branch&Bound analog zum Beispiel aus der Vorlesung, um die optimale Füllung der Hamsterbacken zu bestimmen. Geben Sie jeweils direkt vor Ausführung von $A$=POP($S$), den aktuellen Wert von Bound b, den Wert der unteren Schranke $g(\text{TOP}(S))$, und den Inhalt des Stacks an. Geben sie für $g(\text{TOP}(S))$ an, wie sich dieser Wert zusammensetzt.

Wenn dieser zum Beispiel für die Belegung `101**` durch die optimale mögliche Lösung des fraktionalen greedy Algorithmus abgeschätzt wurde, dann ist $g(\texttt{101**}) = -21.\overline{6} = -(20 + 1/6 * 10)$ anzugeben.

**Beachten sie:** Es gibt Fälle, in denen $A$ nur noch ein Element $x$ enthält, obwohl in der Maske noch Sterne vorkommen, und daher $A = \{x\}$. Das beruecksichtigt der Algorithmus.

[1] Dieses Video dient nur der Veranschaulichung wieviel ein Hamster in sein Maul bekommt: https://www.youtube.com/watch?v=Ms3QdGIzltU.

**Lösung:**

| $b$ | $g(\text{TOP}(S))$ | Stack $S$ |
|---|---|---|
| $\infty$ | -47.5$=-(15+15+5+5/8*20)$ | `*****` |
| $\infty$ | -40$=-(10+15+15)$ | `1****, 0****` |
| $\infty$ | -40$=-(10+15+15)$ | `11***, 10***, 0****` |
| $\infty$ | -40$=-(10+15+15)$ | `110**, 10***, 0****` |
| -30 | -30$=-(10+15+5)$ | `1101*, 1100*, 10***, 0****` |
| -40 | -40$=-(10+15+15)$ | `1100*, 10***, 0****` |
| -40 | -30$=-(10+5+15)$ | `10***, 0****` |
| -40 | -47.5$=-(15+15+5+5/8*20)$ | `0****` |
| -40 | -47.5$=-(15+15+5+5/8*20)$ | `01***, 00***` |
| -40 | -35$=-(15+20)$ | `011**, 010**, 00***` |
| -40 | -35$=-(15+5+15)$ | `010**, 00***` |
| -40 | -35$=-(15+5+6/8*20)$ | `00***` |

Folgende verwendete Masken entsprechen nur einem Element: 1101*, 1100*, 011**, 010**

Die optimale Beladung ist folglich 11001, also { 1: Sellerie, 2: Sonnenblumenkerne, 5: Buchweizen }

Hier nochmal kurz wie man das fraktionale Problem für ***** lösen würde

| Objekt $i$ | Nutzen $u_i$ | Anteil $x_i$ |
|---|---|---|
| 1: Sellerie | 0.016 | 0 |
| 2: Sonnenblumenkerne | 0.15 | 1 |
| 3: Rote Beete | 0.025 | 5/8 |
| 4: Hirse | 0.05 | 1 |
| 5: Buchweizen | 0.075 | 1 |

Optimaler Gesamtgewinn: $15+15+5+5/8*20=47.5$

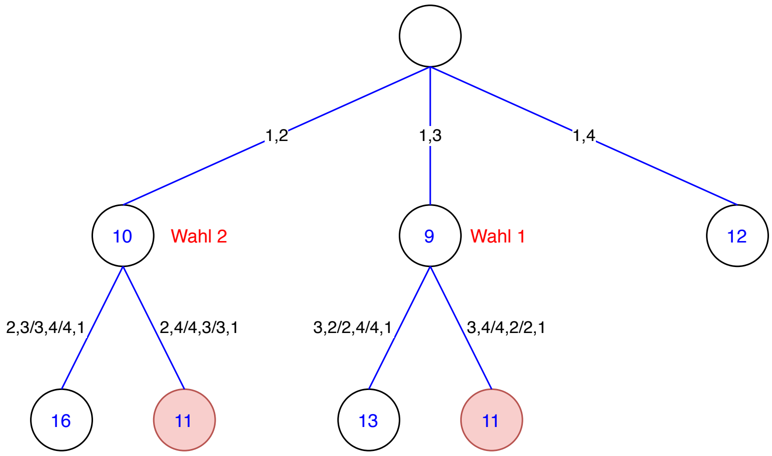
# 1 Branch and Bound Rundreise

Gegeben sei die folgende Entfernungsmatrix:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | - | 5 | 1 | 7 |
| 2 | 5 | - | 2 | 3 |
| 3 | 1 | 2 | - | 2 |
| 4 | 7 | 3 | 2 | - |

Bestimmen Sie unter Verwendung des in der Vorlesung angegebenen Algorithmus die Länge der kürzesten Rundreise $g[1, \{2, ..., n\}]$ mittels **Branch and Bound**. Geben Sie die berechnete Baumstrukture an, sowie Abschätzungen und Distanzen. Geben Sie außerdem alle kürzesten Rundreise an.

**Lösung:**




| $x_{1,2} = 5$ | 1 | 3 | 4 |
|---|---|---|---|
| 2 | - | 2 | 3 |
| 3 | 1 | - | 2 |
| 4 | 7 | 2 | - |

Summe der Zeilen-Minima $2 + 1 + 2 = 5$
$+$ Kosten von $x_{1,2} = 5$ macht 10

| $x_{1,3} = 1$ | 1 | 2 | 4 |
|---|---|---|---|
| 2 | 5 | - | 3 |
| 3 | - | 2 | 2 |
| 4 | 7 | 3 | - |

Summe der Zeilen-Minima $3 + 2 + 3 = 8$
$+$ Kosten von $x_{1,3} = 1$ macht 9

| $x_{1,4} = 7$ | 1 | 2 | 3 |
|---|---|---|---|
| 2 | 5 | - | 2 |
| 3 | 1 | 2 | - |
| 4 | - | 3 | 2 |

Summe der Zeilen-Minima $2 + 1 + 2 = 5$
$+$ Kosten von $x_{1,4} = 7$ macht 12

$$x_{1,3} + [x_{3,2} + x_{2,4} + x_{4,1}] = 1 + 2 + 3 + 7 = 13$$

$$x_{1,3} + [x_{3,4} + x_{4,2} + x_{2,1}] = 1 + 2 + 3 + 5 = 11$$

$$x_{1,2} + [x_{2,3} + x_{3,4} + x_{4,1}] = 5 + 2 + 2 + 7 = 16$$

$$x_{1,2} + [x_{2,4} + x_{4,3} + x_{3,1}] = 5 + 3 + 2 + 1 = 11$$