

Detekování molekul

Petr pracuje ve společnosti, která vyvíjí stroje na detekci molekul. Každá molekula má kladnou celočíselnou váhu. Stroj má *detekční rozsah* $[l, u]$, kde l a u jsou kladná celá čísla. Takový stroj může detekovat určitou množinu molekul právě tehdy, když tato množina obsahuje podmnožinu molekul, jejichž součet vah leží v detekčním rozsahu stroje.

Řečeno formálně, uvažujme n molekul s kladnými celočíselnými vahami w_0, \dots, w_{n-1} . Detekce bude úspěšná, jestliže existuje množina navzájem různých indexů $I = \{i_1, \dots, i_m\}$ taková, že $l \leq w_{i_1} + \dots + w_{i_m} \leq u$.

Je zajištěno, že rozdíl mezi l a u bude vždy větší nebo roven rozdílu mezi vahami nejtěžší a nejlehčí molekuly. Formálně tedy platí, že $u - l \geq w_{max} - w_{min}$, kde $w_{max} = \max(w_0, \dots, w_{n-1})$ a $w_{min} = \min(w_0, \dots, w_{n-1})$.

Vaším úkolem je napsat program, který buď najde jakoukoliv podmnožinu molekul, jejichž součet vah leží v detekčním rozsahu, nebo zjistí, že taková podmnožina neexistuje.

Implementační detaily

Implementujte jednu funkci (metodu):

- `int[] solve(int l, int u, int[] w)`
 - l a u : krajní meze detekčního rozsahu,
 - w : váhy molekul.
 - pokud požadovaná podmnožina existuje, funkce vrátí pole indexů těch molekul, které tvoří jednu takovou vhodnou podmnožinu. Jestliže existuje více správných řešení, nalezněte jedno libovolné z nich.
 - pokud požadovaná podmnožina neexistuje, funkce vrátí prázdné pole.

V programovacím jazyce C vypadá hlavička funkce následovně:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - n : počet prvků v poli w (tj. počet molekul),
 - ostatní parametry mají stejný význam, jako je uvedeno výše.
 - namísto toho, aby funkce vracela pole obsahující m indexů (jako je tomu výše), zde funkce zapíše příslušné indexy do prvních m prvků pole `result` a vrátí hodnotu m .
 - pokud požadovaná podmnožina neexistuje, function nic nezapíše do pole `result` a vrátí hodnotu `0`.

Váš program může zapsat indexy vybraných molekul do návratového pole (nebo do pole 'result' v případě programovacího jazyka C) v libovolném pořadí.

Detaily implementace ve vašem programovacím jazyce najdete v poskytnutých vzorových souborech.

Příklady

Příklad 1

`solve(15, 17, [6, 8, 8, 7])`

V tomto příkladu máme čtyři molekuly s vahami 6, 8, 8 a 7. Stroj může detekovat podmnožiny molekul se součtem vah mezi 15 a 17, včetně obou krajních hodnot. Poznamenejme, že opravdu platí $17 - 15 \geq 8 - 6$. Součet vah molekul číslo 1 a 3 je $w_1 + w_3 = 8 + 7 = 15$, takže funkce může vrátit pole `[1, 3]`. Dalšími možnostmi správné odpovědi jsou pole `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) a `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

Příklad 2

`solve(14, 15, [5, 5, 6, 6])`

V tomto příkladu máme čtyři molekuly s vahami 5, 5, 6 a 6. Hledáme jejich podmnožinu se součtem vah mezi 14 a 15, včetně. Opět si všimněte, že $15 - 14 \geq 6 - 5$. Žádná podmnožina molekul se součtem vah mezi 14 a 15 neexistuje, takže funkce vrátí prázdné pole.

Příklad 3

`solve(10, 20, [15, 17, 16, 18])`

V tomto příkladu máme čtyři molekuly s vahami 15, 17, 16 a 18. Hledáme jejich podmnožinu se součtem vah mezi 10 a 20, včetně. Opět platí $20 - 10 \geq 18 - 15$. Každá podmnožina tvořená právě jednou molekulou má celkovou váhu mezi 10 a 20, takže existují čtyři možné správné odpovědi: `[0]`, `[1]`, `[2]` a `[3]`.

Podúlohy

- (9 bodů): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$, všechny hodnoty w_i jsou shodné.
- (10 bodů): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$,
 $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
- (12 bodů): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$.
- (15 bodů): $1 \leq n \leq 10000$, $1 \leq w_i, u, l \leq 10000$.
- (23 bodů): $1 \leq n \leq 10000$, $1 \leq w_i, u, l \leq 500000$.
- (31 bodů): $1 \leq n \leq 200000$, $1 \leq w_i, u, l < 2^{31}$.

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím tvaru:

- řádek 1: celá čísla n , l , u .
- řádek 2: n celých čísel: w_0, \dots, w_{n-1} .