## Assignment 8.1: Jumps (4 points)

You are playing a game which has $n$ levels. You start from the level $0$ and your goal is to reach the level $n$ by jumping from a level to another. From every level you're able to jump only to $a$ or $b$ levels higher at a time. In how many different ways can you complete the game?

e.g.: let $n = 8$, $a = 2$ and $b = 3$, there are $4$ different ways to pass the game.

- $0 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 8$
- $0 \rightarrow 2 \rightarrow 5 \rightarrow 8$
- $0 \rightarrow 3 \rightarrow 5 \rightarrow 8$
- $0 \rightarrow 3 \rightarrow 6 \rightarrow 8$

Create a function `jumps(n: int, a: int, b: int)` in Python which returns the number of all possible ways to complete the game.

Limist (you can assume that):

- $n \leq 10000$
- $1 \leq a < b \leq n$

Targets:

1. correct solution: $2$ points
2. correct solution $+$ performs in $\Theta(n)$ time: $+2$ points

A code template with an example program:

```
def jumps(n, a, b):
    # TODO


if __name__ == "__main__":
    print(jumps(4, 1, 2))
    print(jumps(8, 2, 3))
    print(jumps(11, 6, 7))
    print(jumps(30, 3, 5))
    print(jumps(100, 4, 5))
```

Output:

```
$ python jumps.py
5
4
0
58
1167937
```

Submit your solution in CodeGrade as `jumps.py`.

## Assignment 8.2: All Sums (4 points)

$A$ is a list consisting of $n$ integers. How many different sums can be generated with the given integers?

For example:

- list $[1, 2, 3]$ has 6 possible sums: $1, 2, 3, 4, 5$ and $6$
- list $[2, 2, 3]$ has 5 possible sums: $2, 3, 4, 5$ and $7$

Create a function `sums(A: list)` in Python which computes the number of all different sums.

Limits (you can assume that): $1 \leq n, a_i \leq 100$

Targets

1. correct solution: $2$ points
2. correct solution $+$ performs at least in $\Theta(n^3)$ time: $+2$ points

A code template with an example program:

```
# sums.py


def sums(items):
    # TODO


if __name__ == "__main__":
    print(sums([1, 2, 3]))
    print(sums([2, 2, 3]))
    print(sums([1, 3, 5, 1, 3, 5]))
    print(sums([1, 15, 5, 23, 100, 55, 2]))
```

```
5
18
121
```

Submit your solution in CodeGrade as `sums.py`.

Last modified: Wednesday, 31 July 2024, 5:34 PM