

## Øving 4, Algoritmer og Datastrukturer

### Del 1

Den ytre while-løkken kjører  $n$  ganger, ettersom den kjører hver gang en person blir drept.

Den indre for-løkken referer til det hvor mange personer man hopper over før en node tar selvmord og kjører  $m$  ganger.

Da dette er en nøstet løkke må verdiene multipliseres, som resulterer i en øvre grensen for tidskompleksiteten for programmet gitt av  $O(m*n)$ .

Nedre grense er konstant,  $\Omega(1)$ , fordi når  $m = 1$  vil metoden returnere  $m$  helt i begynnelsen av programmet.

```
while (current.getNext() ≠ current){
    for (int i = 0; i < mInterval; i++) {
        previous = current;
        current = current.getNext();
    }
    previous.removeNext();
}
```

### Del 2

Svar på oppgave 2 finnes i filen BracketSolver.java. Klassen kan kompileres og lese en tekstfil ved bruk av det første argumentet til programmet:

Eksempel på kjøring:

```
PS D:\AlgDat\tasks\task4\part1\task4copy\src> javac BracketSolver.java
PS D:\AlgDat\tasks\task4\part1\task4copy\src> java BracketSolver testData.txt
{[]() ) : false
{([() ]}] : false
{ } : false
[{} ] : false
(){}{[[]][ ]}{() } : true
```

Eksempel på fil som kjøres:

```
1  {[]()  
2  {[()]})  
3  (  
4  [{}]  
5  (){{}[[]][[]]}{()  
6
```

Om det finnes andre karakterer enn parenteser vil disse bli fjernet, og skal ikke påvirke resultatet på programmet