

Efficient Algorithms

Task 03 - Josphus Problem

Code Josephus

```
def josephus(n,k):  
    r = 1  
    for i in range(1, n + 1):  
        r = (r + k) % i  
    return r + 1
```

Ergebnis

Brute force

Innerhalb von 10 Minuten kann man alle Überlebenden bis zu einer Kreisgröße von 103.921 Personen ermitteln

Eine Stelle in 10 Minuten

Innerhalb von 10 Minuten kann man den Überlebenden einer Kreisgröße von 1.350.000.000 Personen ermitteln.

Documented Code

```

from multiprocessing import Process
import time

MAX_N      = 1000000
K          = 2
TEN_MINUTES = 600

def run_with_limited_time(func, args, kwargs, time):
    """Runs a function with a time limit

        :param      func:      (function) the function to run
                                e.g.
                                josephus
        :param      args:      (Tuple) The functions arguments
                                e.g.
                                (K,MAX_N)
        :param      kwargs:    (dictionary) The functions keywords
                                e.g.
                                {}
        :param      time:      (int) The time limit in seconds
                                e.g.
                                600 - runs as long as the function does not take longer than 600 seconds

        :return     boolean:    if the function ended successfully
                                e.g.
                                True - if ended successfully
                                or
                                False - if ended unsuccessfully
    """
    p = Process(target=func, args=args, kwargs=kwargs)
    p.start()
    p.join(time)
    if p.is_alive():
        p.terminate()
        return False

    return True

```

```

def josephus(n, k):
    """ Calculate the survivor of a josephus circle

        :param      n:  (int) length of the josephus circle
                           e.g.
                           2 - to have a josephus circle with 2 people
                           or
                           10.000 - to have a josephus circle with 10.000 people
        :param      k:  (int) number of steps
                           e.g.
                           2 - to kill every second person in the josephus circle
                           or
                           3 - to kill every third person in the josephus circle

        :return:  r+1:  (int) number of the survivor
                           e.g.
                           3 - if the length n = 3  and k = 2
                           or
                           5 - if the length n = 10 and k = 2

        r+1 because we can not get the modulo of number 0
    """

    # (int) number of the survivor
    r = 1

    # for index i in list (range(1,n+1) --> creates a list)
    # e.g. n = 3 --> range(1,4) = [1,2,3]
    for i in range(1,n+1):

        # (int) number of the survivor
        # e.g.
        # first iteration: r = 0, k = 2, i = 1 --> (0+2)%1 = 0
        # second iteration: r = 0, k = 2, i = 2 --> (0+2)%2 = 0
        # last iteration: r = 0, k = 2, i = 3 --> (0+2)%3 = 2
        r = (r+k)%i

    # return the survivor
    # e.g.
    # r == 2 --> r+1 = 3

    print "*****"
    print r+1
    print "*****"
    print "\n"

    return r+1

```

```

def bf_jos(func, args, kwargs):
    """Runs a function that has to be brute forced

    :param func: (function) the function to run
                  e.g.
                  josephus
    :param args: (Tuple) The functions arguments
                  e.g.
                  (K, MAX_N)
    :param kwargs: (dictionary) The functions keywords
                   e.g.
                   {}

    :return boolean: if the function ended successfully
                     e.g.
                     True - if ended successfully
    """
    a = list(args)

    for i in range(1, a[1]):

        arguments = list()
        arguments.insert(0, i)
        arguments.insert(1, a[0])
        arguments = tuple(arguments)

        print "#####"
        print "# " + str(i) + " #"
        print "#####"

        p = Process(target=func, args=arguments, kwargs=kwargs)
        p.start()
    return True

```

```
# Main program
if __name__ == '__main__':

    # if we brute force the josephus function
    # we will get to number 103.921 within 10 minutes
    # if we only calculate one number within 10 minutes
    # we are able to get the survivor of 1.350.000.000 Persons

    run_with_limited_time(bf_jos, (josephus, (K,MAX_N), {}), {}, TEN_MINUTES)

    #start_time = time.time()
    #print "1350000000"
    #print josephus(1350000000, K)
    #print("--- %s seconds ---" % (time.time() - start_time))

    #print "103921"
    #print josephus(103921,K)
```