Heimadæmi – heimadæmi 2

Arnar Sigurðsson

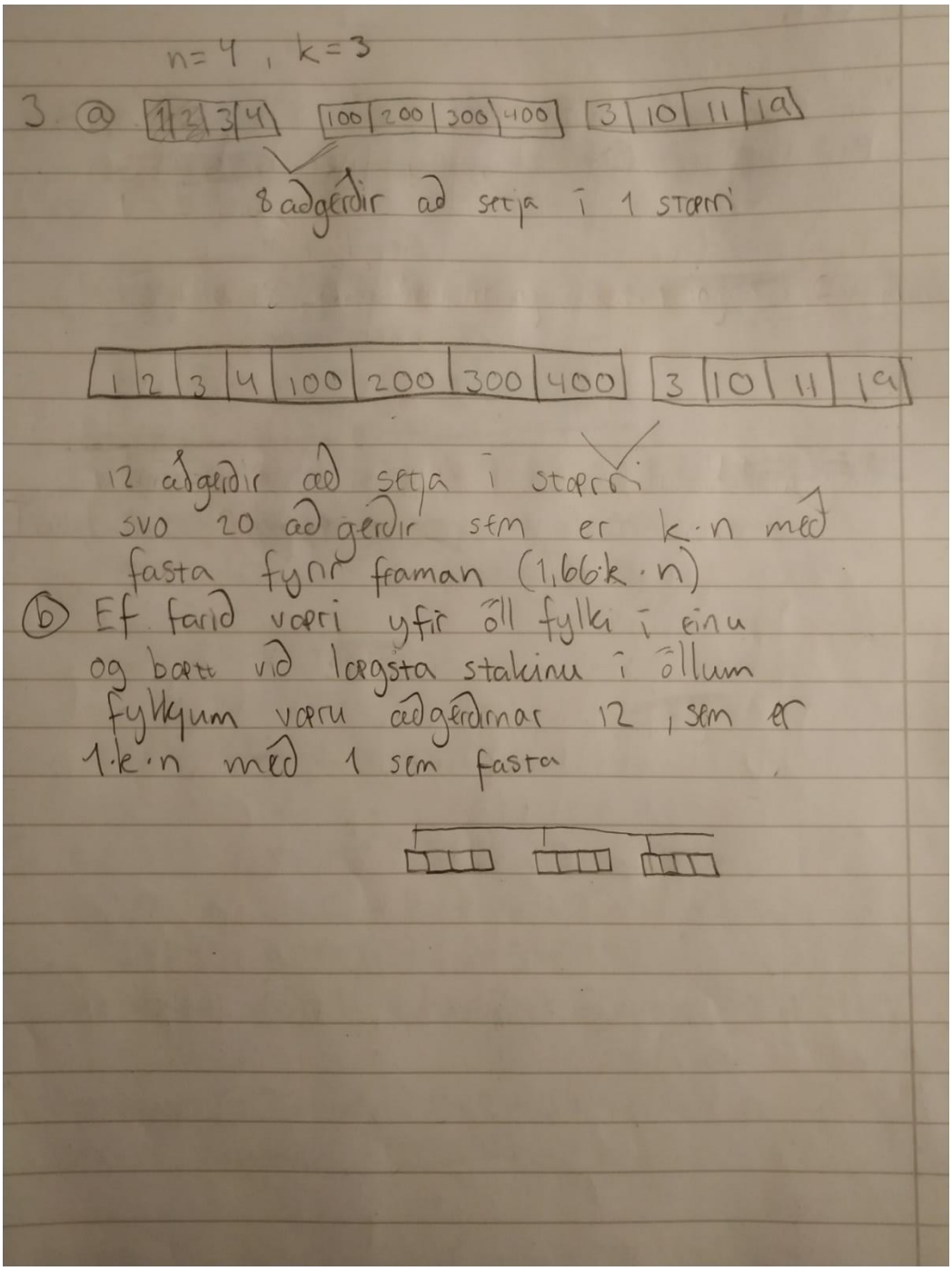
1.

```python
import math

def peasantIterative(x, y):
    prod = 0
    while x > 0:
        if x % 2 != 0:
            prod = prod + y
        x = math.floor(x/2)
        y = y + y
    return prod

def peasantRecursive(x, y):
    if x == 0:
        return 0
    else:
        xNew = math.floor(x/2)
        yNew = y + y
        prod = peasantRecursive(xNew, yNew)
        if x % 2 != 0:
            prod = prod + y
        return prod

print(peasantIterative(1501, 2022))
print(peasantRecursive(1501, 2022))
```

```
C:\Users\addi\Desktop\Háskóli\onn 4\GreiningReiknirita\Vika2>daemi1.py
3035022
3035022
```

2.

3.



4. a) Vegna þess að O() segir til um hvernig reiknirit hagar sér þegar inntakið n nálgast óendanlegt, og í þessu tilfelli verður listinn sem insertion sort þarf að raða alltaf meira og meira í réttri röð eftir því sem n stækkar, því quicksort er búið að setja fullt af stökum í rétta röð, og insertion sort er O(n) þegar listinn er þegar raðaður.

   b) Vegna þess að það kostar að framkvæma endurkvæmni og kalla á föll og þetta „overhead“ getur valdið því að á einhverjum tímapunkti verður hagkvæmara að skipta yfir í „dýrara“ reiknirit eins og insertion sort þegar inntakið er nógu lítið.

5.

```python
# O(1) skv. fyrirmælum
def flip(listi, k):
    left = k
    right = len(listi) - 1
    while left < right:
        temp = listi[left]
        listi[left] = listi[right]
        listi[right] = temp
        right = right - 1
        left = left + 1

#stærsta pönnukaka = 0, minnsta pönnukaka = lengd á lista - 1
def pancakeFlip(listi):
    org = listi.copy()
    finished = 0
    count = 0
    while  finished < len(listi):
        curr = finished

        # leit að stærstu pönnuköku er hugsað sem O(1) skv. fyrirmælum
        while curr < len(listi):
            if listi[curr] == finished:
                if curr != finished:
                    if curr != len(listi) -1:
                        flip(listi, curr)
                        count = count + 1
                    if listi[finished] != finished:
                        flip(listi, finished)
                    count = count + 1
                finished = finished + 1
                break
            curr = curr + 1

    print("upphaflegi staflinn: ", org, "| raðaðar pönnukökur: ", listi, " og fjöldi snúninga sem þurfti:", count)

pancakeFlip([0, 1, 2])
pancakeFlip([0, 2, 1])
pancakeFlip([1, 0, 2])
pancakeFlip([1, 2, 0])
pancakeFlip([2, 1, 0])
pancakeFlip([2, 0, 1])
```

```
C:\Users\addi\Desktop\Háskóli\onn 4\GreiningReiknirita\vika2>pancakes.py
upphaflegi staflinn:  [0, 1, 2] | raðaðar pönnukökur:  [0, 1, 2]  og fjöldi snúninga sem þurfti: 0
upphaflegi staflinn:  [0, 2, 1] | raðaðar pönnukökur:  [0, 1, 2]  og fjöldi snúninga sem þurfti: 1
upphaflegi staflinn:  [1, 0, 2] | raðaðar pönnukökur:  [0, 1, 2]  og fjöldi snúninga sem þurfti: 3
upphaflegi staflinn:  [1, 2, 0] | raðaðar pönnukökur:  [0, 1, 2]  og fjöldi snúninga sem þurfti: 2
upphaflegi staflinn:  [2, 1, 0] | raðaðar pönnukökur:  [0, 1, 2]  og fjöldi snúninga sem þurfti: 1
upphaflegi staflinn:  [2, 0, 1] | raðaðar pönnukökur:  [0, 1, 2]  og fjöldi snúninga sem þurfti: 2
```

Versti fjöldi snúninga er breytilegur, í n=3 er það 3 en í n=1 er það 0 sem dæmi og í n=4 er það 5 svo jafnan væri 2n-3.