

a)

ATTTTCATTATCAGGA

ATTT

TTTT

TTTC

TTCA

TCAT

CATT

ATTA

TTAT

TATC

ATCA

TCAG

CAGG

AGGA

Iz tega niza lahko dobimo 91 podzaporedij, ki se tudi ujemajo:

13x po 1 odčitek

12x po 2 odčitka

11x po 3 odčitke

10x po 4 odčitke

9x po 5 odčitkov

8x po 6 odčitkov

7x po 7 odčitkov

6x po 8 odčitkov

5x po 9 odčitkov

4x po 10 odčitkov

3x po 11 odčitkov

2x po 12 odčitkov

1x po 13 odčitkov

=====

SUM: 92 kombinacij odčitkov

SUM = 0

for i in range(1:len(niz)-k+1):

SUM += len(niz)-k+1 - i + 1

Niz je enolično sestavljen, tako da ne morem odčitke menjati in s tem kreirati še več zapisov

b)

Vozlišča se ne smejo ponavljati -> `len(set(vozlisca)) == len(vozlisca)`

Število vhodnih povezav in izhodnih povezav v in iz vozlišča mora biti 1. Če je več vhodnih povezav, potem se vozlišča ponavljajo, prav tako v tem vozlišču nebi vedli v katero vejo se razvejati

naprej (več možnih)

Graf mora biti povezan, tako da lahko v vsako vozlišče pridemo iz nekega vozlišča
Zagotoviti moramo da gre za eulerjev sprehod (če sta največ 2 vozlišči semi-uravnoteženi in vsi ostali uravnoteženi). Prav tako mora obstajati hamiltonov sprehod.

c)

Algoritem sestavlja odčitke skupaj kjer se ujemajo zadnja in prva črka. Ker je teh ujemanj lahko več,

grem zato s for zanko skozi in se potem razvejam z rekurzijo in prečerkiram še druga vozlišča od tega naprej

Če pridem do konca, sem uspešno sestavil niz in potem niz vrnem.

Dela na podoben način kot permutacije, vendar še preverjam ujemanja črk.

Algoritem ima lahko enaka vozlišča, zato se tudi razveja z rekurzijo in ni linearen.

Algoritem:

```
def kreiraj_vozlisca(odcitki, LEN_ODCITEK):
    vozlisca = []
    vozlisca.append(odcitki[0][:LEN_ODCITEK-1])
    for i in range(len(odcitki)):
        vozlisca.append((odcitki[i][1: LEN_ODCITEK]))
    return vozlisca

def sestavi_vozlisca(vozlisca, i):
    #rekurzivno sestavljam vozlišča ki pašejo skupaj. če pridemo do konca-
    i>=len(vozlisca) potem se vrnemo
    #gremo pa skozi vsa vozlišča, saj je lahko več vozlišč istih, in se moramo razvejati
    z rekurzijo
    if i >= len(vozlisca):
        return True

    tmp_vozlisce = vozlisca[i]
    print(tmp_vozlisce)
    for vozlisce in vozlisca:
        if(tmp_vozlisce[1:] == vozlisce[:len(vozlisce)-1]): # ce mogoce pašeta
            skupej... pol se pomakn naprej
            sestavi_vozlisca(vozlisca, i + 1)
            #if():

def main():
    LEN_ODCITEK = 4
    odcitki = ["AGGA", "ATCA", "ATTA", "ATTT", "CAGG", "CATT", "TATC", "TCAG", "TCAT",
               "TTAT", "TTCA", "TTTC", "TTTT"]

    vozlisca = kreiraj_vozlisca(odcitki, LEN_ODCITEK)
    print(vozlisca)

    for i in range(len(vozlisca)):
        sestavi_vozlisca(vozlisca, i)
        print("")

main()
```