Algorytmy i struktury danych, Teleinformatyka, I rok

Raport z laboratorium nr: 1

Imię i nazwisko studenta: Filip Komarzyniec

nr indeksu: 296913

1. W pole poniżej wklej najważniejszy (według Ciebie) fragment kodu źródłowego z zajęć (maksymalnie 15 linii).

```
1. def Move_disk(sor, dest) :
2.         global count
3.         if count == 0 :
4.                 dest.append(sor.pop())
5.                 tsor = tuple(sor)
6.                 tdest = tuple(dest)
7.                 dict = { tuple(SOR) : 1 , tuple(DEST) : 3, tuple(BUFF) : 2 }
8.                 print("moving from : ",dict[tsor], "to : ", dict[tdest] , sep='\t ')
9.         else :
10.                tsor = tuple(sor)
11.                tdest = tuple(dest)
12.                dict = { tuple(SOR) : 1 , tuple(DEST) : 3, tuple(BUFF) : 2 }
13.                print("moving from : ",dict[tsor], "to : ", dict[tdest] , sep='\t ')
14.                dest.append(sor.pop())
15.        count+=1
```

Uzasadnij swój wybór.

Powyższy fragment kodu jest całą funkcją przenoszącą krążek w implementacji rekurencyjnej problemu wież Hanoi, wypisującą dodatkowo poszczególne ruchu dysków. Jest ona krótsza, bardziej przejrzysta oraz łatwiejsza do zastosowania niż odpowiadająca jej funkcja w implementacji iteracyjnej badanego problemu.

2. Podsumuj wyniki uzyskane podczas wykonywania ćwiczenia. Jeśli instrukcja zawierała pytania, odpowiedz na nie.

Z przeprowadzonych przeze mnie prób można wyciągnąć wniosek, iż dla większej liczby krążków algorytm rekurencyjny jest szybszy. Wyniki prezentują się następująco :

	rekurencja		iteracja	
7 krążków	0.04465	[s]	0.043645	[s]
17 krążków	2.52009	[s]	2.9857	[s]
21 krążków	40.86877	[s]	50.02913	[s]

Algorytm rekurencyjny jest bardziej optymalny pod względem szybkości działania. Biorąc zaś pod uwagę ilość potrzebnych zasobów jest on mniej optymalny.

Raporty należy wysłać na adres: andmat+aisd@agh.edu.pl . Termin oddania raportów to 7 dni po zajęciach laboratoryjnych, których dotyczy raport.