

# 알고리즘 실습 보고서

- Knapsack & Sequence -

전공 : 컴퓨터공학과

분반 : 05반

학번 : 201502085

이름 : 이규봉

## 1. 실행 환경

과제의 코드를 테스트한 환경은 아래와 같습니다. 따로 사용한 라이브러리는 없습니다.

- Windows 10
- Pycharm 2019.02
- Python 3.7.4

## 2. 과제 설명

- Knapsack & Sequence

## 3. 문제 해결 방법

- 문제를 해결하기 위해 자신이 사용한 방법, 아이디어에 대한 설명  
(구현한 코드에 대한 설명 등)

1 -

knapback problem의 경우, 학습 목적으로 원래 C++로 풀었던 방법대로 풀어 pack이란 함수로 만들었습니다. 그리고 ppt에 나온 의사코드대로 다시 풀어 pack\_2로 풀었습니다. pack\_2는 DP를 이용하여 pack\_2.opt란 배열을 opt로 구하는 함수입니다. 그 외 별개의 analysis 함수에선 pack\_2로 구한 opt 배열을 순회하며 어떤 아이템을 꺼낸 것인지를 찾아 출력합니다.

2 -

sequence alignment 문제의 경우, 의사코드 대로 코드를 짰 후 해당 값들을 출력하는 형식으로 확인했습니다. 해당 코드로 구한 표는 아래와 같습니다.

서열	X			t		c		a		t		a		g		t		t		a		a		c		a
Y	index	0		1		2		3		4		5		6		7		8		9		10		11		12
	0	0	-2	-2	-4	-4	-6	-6	-8	-8	-10	-10	-12	-12	-14	-14	-16	-16	-18	-18	-20	-20	-22	-22	-24	-24
		-2	1	-4	-3	-6	-5	-8	-5	-10	-9	-12	-11	-14	-11	-16	-13	-18	-17	-20	-19	-22	-21	-24	-23	-26
t	1	-2	-4	1	-1	-1	-3	-3	-5	-5	-7	-7	-9	-9	-11	-11	-13	-13	-15	-15	-17	-17	-19	-19	-21	-21
		-4	-3	-1	2	-3	-2	-5	-4	-7	-6	-9	-8	-11	-10	-13	-12	-15	-14	-17	-16	-19	-16	-21	-20	-23
c	2	-4	-6	-1	-3	2	0	0	-2	-2	-4	-4	-6	-6	-8	-8	-10	-10	-12	-12	-14	-14	-16	-16	-18	-18
		-6	-5	-3	-2	0	3	-2	-1	-4	-1	-6	-5	-8	-7	-10	-9	-12	-9	-14	-11	-16	-15	-18	-15	-20
a	3	-6	-8	-3	-5	0	-2	3	1	1	-1	-1	-3	-3	-5	-5	-7	-7	-9	-9	-11	-11	-13	-13	-15	-15
		-8	-7	-5	-4	-2	-1	1	2	-1	0	-3	0	-5	-4	-7	-6	-9	-8	-11	-10	-13	-12	-15	-14	-17
g	4	-8	-10	-5	-7	-2	-4	1	-1	2	0	0	-2	0	-2	-2	-4	-4	-6	-6	-8	-8	-10	-10	-12	-12
		-10	-9	-7	-6	-4	-1	-1	0	0	3	-2	-1	-2	-1	-4	-3	-6	-3	-8	-5	-10	-9	-12	-9	-14
a	5	-10	-12	-7	-9	-4	-6	-1	-3	0	-2	3	1	1	-1	-1	-3	-3	-5	-3	-5	-5	-7	-7	-9	-9
		-12	-11	-9	-8	-6	-3	-3	-2	-2	1	1	2	-1	0	-3	-2	-5	-2	-5	-2	-7	-6	-9	-6	-11
a	6	-12	-14	-9	-11	-6	-8	-3	-5	-2	-4	1	-1	2	0	0	-2	-2	-4	-2	-4	-2	-4	-4	-6	-6
		-14	-13	-11	-10	-8	-7	-5	-4	-4	-3	-1	2	0	1	-2	-1	-4	-3	-4	-3	-4	-3	-6	-5	-8
g	7	-14	-16	-11	-13	-8	-10	-5	-7	-4	-6	-1	-3	2	0	1	-1	-1	-3	-3	-5	-3	-5	-3	-5	-5
		-16	-13	-13	-12	-10	-9	-7	-4	-6	-5	-3	-2	0	3	-1	2	-3	-2	-5	-4	-5	-4	-5	-4	-7
t	8	-16	-18	-13	-15	-10	-12	-7	-9	-4	-6	-3	-5	0	-2	3	1	2	0	0	-2	-2	-4	-4	-6	-4
		-18	-17	-15	-14	-12	-9	-9	-8	-6	-3	-5	-4	-2	-1	1	2	0	3	-2	1	-4	-3	-6	-3	-6
a	9	-18	-20	-15	-17	-12	-14	-9	-11	-6	-8	-3	-5	-2	-4	1	-1	2	0	3	1	-1	-1	-1	-3	-3
		-20	-19	-17	-14	-14	-13	-11	-10	-8	-7	-5	-4	-4	-3	-1	0	0	1	1	2	-1	2	-3	-2	-5
c	10	-20	-22	-17	-19	-14	-16	-11	-13	-8	-10	-5	-7	-4	-6	-1	-3	0	-2	1	-1	2	0	2	0	0
		-22	-21	-19	-16	-16	-15	-13	-12	-10	-9	-7	-6	-6	-5	-3	-2	-2	-1	-1	0	0	3	0	1	-2
c	11	-22	-24	-19	-21	-16	-18	-13	-15	-10	-12	-7	-9	-6	-8	-3	-5	-2	-4	-1	-3	0	-2	3	1	1

x : tcataagtaaca / y : tcagaagtacc

## 4. 결과 화면

```
C:\Users\jopem\Anaconda3\python.exe C:/Algorithm-HW/Assign08/knapback.py
배낭의 사이즈를 입력하세요 (0~50) :
11
0  1  1  1  1  1  1  1  1  1  1  1
0  1  6  7  7  7  7  7  7  7  7  7
0  1  6  7  7  18  19  24  25  25  25  25
0  1  6  7  7  18  22  24  28  29  29  40
0  1  6  7  7  18  22  28  29  34  35  40
max : 40
[3, 4]

Process finished with exit code 0
```

## 5. 느낀점 및 고찰

knapback 문제 자체는 쉬웠지만, knapback 문제에서 어떤 item을 꺼냈는지 확인하는 함수를 짜는 것이 생각보다 힘들었습니다.