

Assignment #2

Turtle Charter 2

Digital Computer Concept and Practice
dccp@hcil.snu.ac.kr

1 Introduction

이번 과제는 Turtle Library를 이용하여 간단한 Treemap을 그려보는 과제입니다.

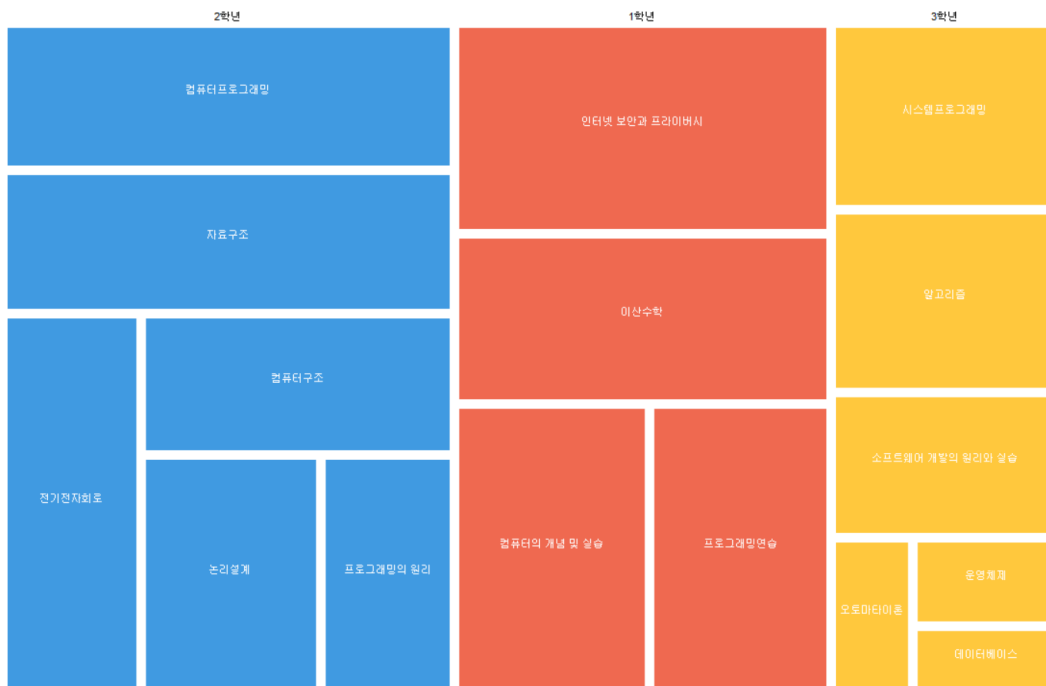


그림 1: 과제 최종 형태

- 제출 기한: 2020년 10월 23일 금요일 **23:59 (자정 전까지 인정)**
- 제출 방법:
 - eTL 과제란에 **소스 코드**(treemap.py 파일)와 **Turtle Treemap 캡처본**을 제출
 - 모든 제출 파일을 **학번-이름** 형태의 **압축 파일 (zip)** 형식으로 묶어 제출해주시길 바랍니다.
 - * 2020-12345-홍길동.zip
 - treemap.py - 작성한 소스 코드
 - cse.jpg - DATA_CSE를 그린 결과
 - youtube.jpg - DATA_YOUTUBE를 그린 결과
 - 과제 1과 다르게 소스코드의 파일명은 바꾸지 않고 제출합니다.
 - 이미지의 파일 포맷 형식은 관계 없습니다.

2 Explanation

2.1 Background

Treemap은 여러 카테고리에 속한 항목들의 상대적인 크기를 영역의 넓이에 대응시켜 나타내는 차트입니다. 특히, 큰 카테고리 안에 작은 카테고리가 있고, 작은 카테고리가 더 작은 카테고리로 나뉘는 계층적(hierarchical) 데이터를 보여주기 적합합니다. 하지만 이번 과제에서는 두 단계 이상으로 깊은 카테고리는 고려하지 않고, 여러 항목들이 한 단계의 카테고리에 의해 나뉘지는 경우만을 대상으로 합니다.

본 과제에서는 아래의 예시 그림을 보면, 크게 아침과 점심이라는 카테고리가 있고 각 카테고리에 속한 여러 항목이 있습니다. 예를 들어, 각 항목에 판매량이 대응되어 있다고 할 때, 사각형의 상대적인 넓이는 판매량의 상대적인 비율을 반영합니다.

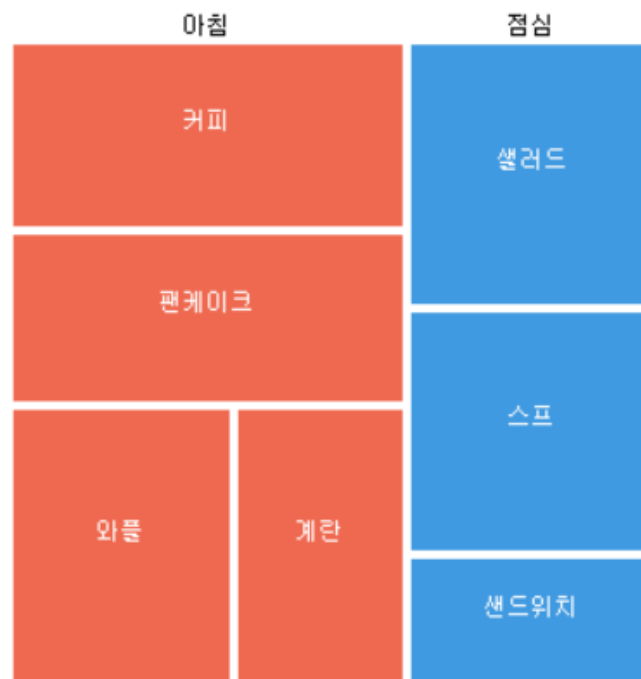


그림 2: Treemap 예시

Treemap은 2차원 상에 그려지는 차트이기 때문에 여러 방식으로 영역을 분할할 수 있으며, 개중에는 복잡하지만 결과적으로 더 보기 쉽고 예쁜 차트를 만드는 방법도 있습니다. 이번 과제에서는 아래와 같은 방법으로 차트를 그립니다.

1. treemap을 그려야 할 전체 캔버스의 직사각형의 좌표가 주어집니다.
2. 우선 각 카테고리 별로 영역을 가로로만 분할합니다. 각 카테고리에 해당하는 영역의 넓이는 카테고리에 속한 항목의 값의 합에 대응되어야 합니다. 또, 영역이 넓은 순으로 왼쪽부터 내림차순으로 정렬되어야 합니다.
3. 각 카테고리 영역 안에서 다음과 같은 과정을 반복합니다.
 - a) 해당 카테고리에 속한 항목 중에서 가장 값이 큰 것을 고릅니다. 단, 이미 화면에 그린 항목은 고르지 않습니다.
 - b) 카테고리 영역 중 무엇인가 그려지지 않고 남은 공간(맨 처음에는 전체 카테고리 영역)을 찾습니다.

- c) 남은 공간의 너비(width)가 높이(height)보다 길거나 같다면 영역을 좌우로 분할해서 왼쪽 영역에 현재 항목을 그립니다. 새로이 그려지는 왼쪽 영역의 넓이(L)와 남은 오른쪽 영역의 넓이(R)의 비율은 현재 항목의 값(L')과 해당 카테고리 내에서 아직 그리지 않은 항목의 값의 합(R')의 비율과 같아야 합니다. (i.e. $L : R = L' : R'$)
- d) 남은 공간의 높이가 더 길다면 영역을 상하로 분할해서 위쪽 영역에 현재 항목을 그립니다. 그려지는 영역과 남은 영역 사이 넓이 비율은 좌우의 경우 (c) 와 같습니다.
- e) 나머지 항목을 남은 영역 안에서 동일한 방법으로 그려나갑니다. 항목이 단 하나만 남았다면 남은 영역 전체에 그립니다. 이전까지 정상적으로 그렸다면 남은 영역의 넓이가 해당 항목의 값과 대응될 것입니다.



그림 3: 남은 공간의 너비가 높이보다 길거나 같은 경우 알고리즘이 동작하는 과정. 더 자세한 설명은 맨 아래의 부록을 참조하세요.

2.2 Implementation

- 제공해드리는 skeleton code 파일은 데이터 `data.py`와 한 개의 소스코드 `treemap.py`로 구성되어 있습니다. 과제 1 번에서 실행과 관련된 어려움이 많았기 때문에 뼈대코드를 최대한 간소화하였습니다.
- `data.py` 안에는 테스트 용도로 제공해드리는 두 개의 데이터가 있습니다. `treemap.py`의 최하단 main 블록에서 그릴 데이터를 선택할 수 있습니다.
- 구현해야 할 함수는 `treemap.py`안에 있는 `def treemap(box_range, data)`입니다.
 - `box_range` : (`x1`, `x2`, `y1`, `y2`)의 형식으로 길이가 4인 tuple이 주어집니다. turtle의 기본 좌표계를 기준으로 해당 영역에 treemap을 그려야 합니다. $x1 < x2$, $y1 < y2$ 임이 보장됩니다.
 - `data` : 리스트가 주어집니다. 각 원소는 하나의 카테고리를 설명하는 튜플입니다.
 - * 카테고리를 설명하는 튜플의 첫 번째 원소는 str 형식이며 해당 카테고리의 이름입니다.
 - * 튜플의 두 번째 이후 원소는 (str, int, str) 형식이며 첫 번째는 항목 이름, 두 번째는 항목의 값, 세 번째는 사각형의 color string입니다.
- 트리맵은 다음과 같은 조건을 만족해야 합니다.

- 서로 다른 사각형 영역은 간격을 통해 구분되어야 합니다. 예를 들어, 동일한 색상의 사각형을 간격도 없이 이어서 그리면 두 사각형을 구분할 수 없게 됩니다. 이를 위해 실제로 사각형을 그릴 영역의 전체를 채우지 말고 영역 안쪽에 5만큼의 여백(padding)을 두어 그려주십시오.
- 직사각형 영역의 정가운데에 항목 이름을 하얀색으로 써주세요. `turtle.write` 함수를 찾아 보고 사용하면 됩니다.
- 카테고리 영역의 위쪽 정가운데에 카테고리 이름을 검은색으로 써주세요. 카테고리 이름은 `box_range`의 바깥에 있어도 괜찮습니다.
- 폰트의 종류와 크기 등은 설정하지 않고 default를 사용하시면 됩니다.
- 텍스트가 사각형 영역을 벗어나는 경우 등은 따로 고려하지 않아도 됩니다. 채점할 때 `box_range`가 충분히 크게 주어질 것입니다.

3 Cautions

- Grading Policy

- 채점할 때는 주어진 뼈대코드에 없는 새로운 데이터를 사용합니다.
- 텍스트의 크기와 위치는 약간의 오차를 허용하지만, 그 외에는 스펙에 주어진 대로 정확히 동작해야 합니다.

- Late Policy

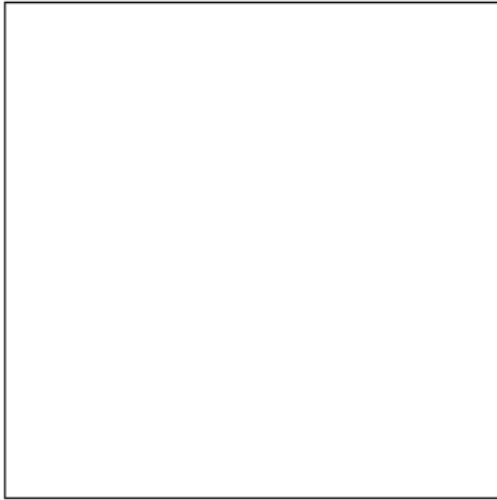
- 연장 제출은 **5일**까지 허용됩니다. (10/28 23:59까지)
- 연장 제출시 하루 **20%**씩 감점됩니다. (10/24 제출 20% 감점, 10/25 제출 40% 감점 ...)

- python 3.8.5 인터프리터로 소스코드가 실행되지 않을 시 **0점 처리**됩니다.
- 문의사항은 eTL 질의응답 게시판 혹은 [조교 메일](#)로 보내주시기 바랍니다.

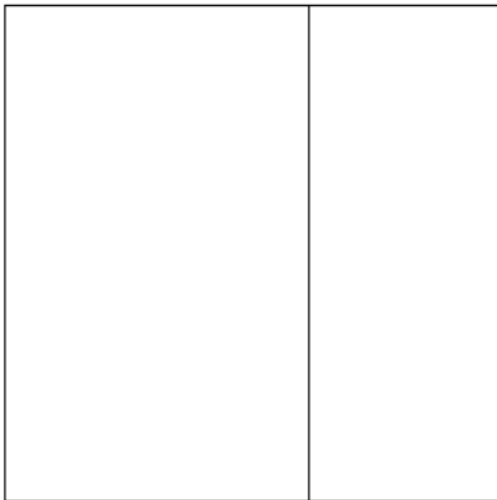
4 References

- [Wikipedia](#)
- [Treemap](#)

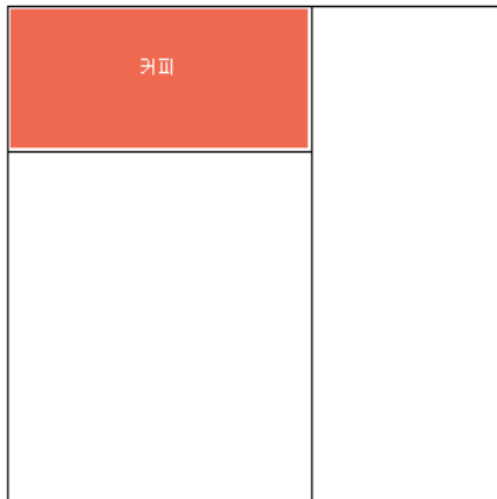
5 Appendix: 알고리즘 설명



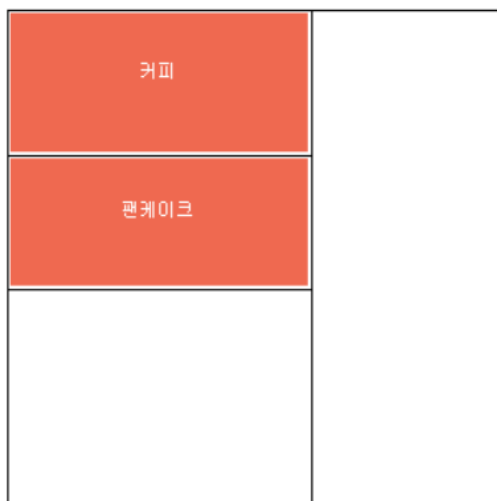
- 이곳이 treemap을 그릴 전체 영역입니다
- 예시 데이터는 다음과 같습니다.
 - 아침: [(와플, 38), (계란, 29), (팬케이크, 41), (커피, 45)]
 - 점심: [(샐러드, 40), (샌드위치, 20), (스프, 37)]



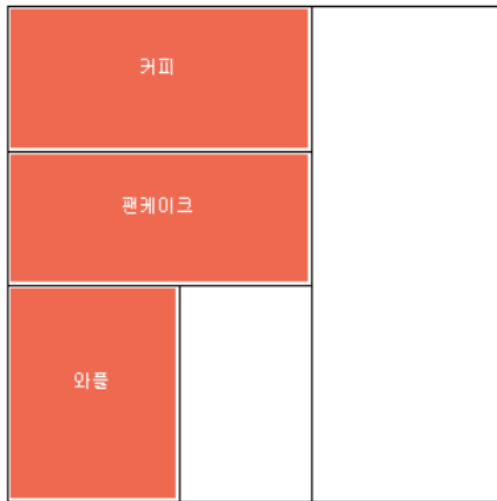
- 아침에 속한 항목값의 총합은 $38 + 29 + 41 + 45 = 153$
- 점심에 속한 항목값의 총합은 $40 + 20 + 37 = 97$
- 내림차순으로 정렬하면 아침이 왼쪽, 점심이 오른쪽에 와야 합니다
- 153:97의 비율로 영역을 나눕니다. 카테고리의 무조건 좌우로 분할합니다.



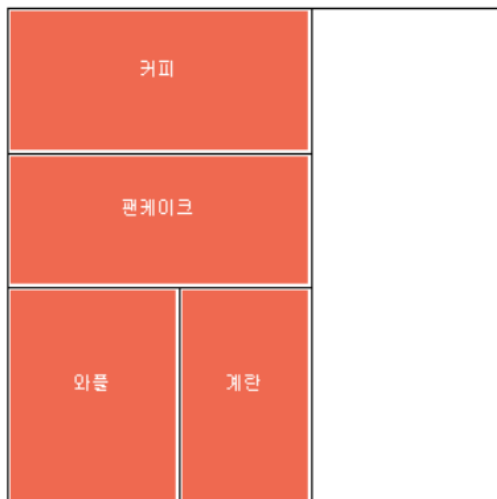
- 아침: [(와플, 38), (계란, 29), (팬케이크, 41), (커피, 45)]
- 제일 큰 것은 (커피, 45)이고, 나머지의 합은 $38 + 29 + 41 = 108$
- 영역의 높이가 너비보다 크기 때문에 45:108 상하 분할



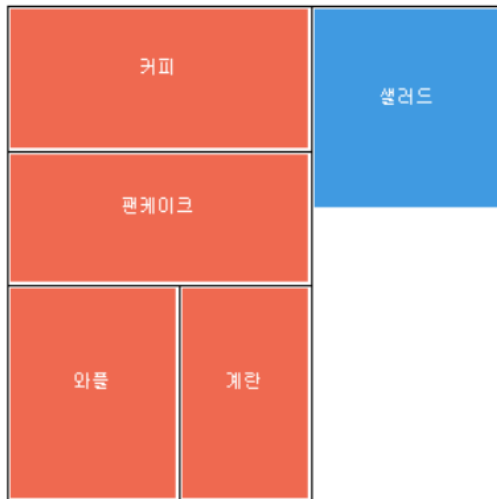
- 아침: [(와플, 38), (계란, 29), (팬케이크, 41)]
- 제일 큰 것은 (팬케이크, 41)이고, 나머지의 합은 $38 + 29 = 67$
- 영역의 높이가 더 크기 때문에 41:67 상하 분할



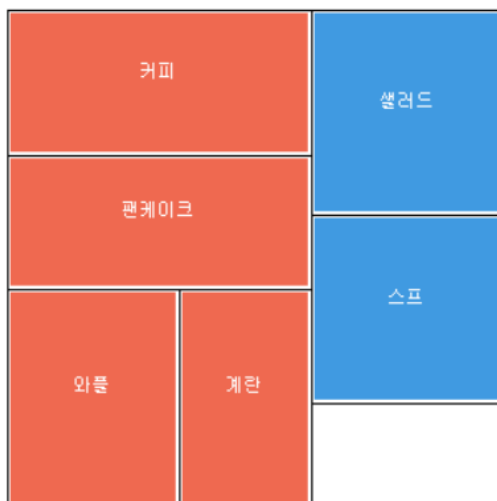
- 아침: [(와플, 38), (계란, 29)]
- 제일 큰 것은 (와플, 38)이고, 나머지의 합은 29
- 영역의 너비가 더 크기 때문에 38:29 좌우 분할



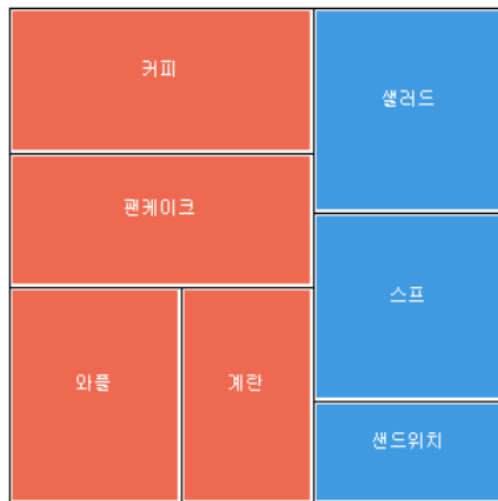
- 아침: [(계란, 29)]
- 항목이 한 개 남았으므로 남은 영역에 채워 넣습니다



- 점심: [(샐러드, 40), (샌드위치, 20), (스프, 37)]
- 제일 큰 것은 (샐러드, 40)이고, 나머지의 합은 $20 + 37 = 57$
- 영역의 높이가 더 크기 때문에 40:57 상하 분할



- 점심: [(샌드위치, 20), (스프, 37)]
- 제일 큰 것은 (스프, 37)이고, 나머지의 합은 20
- 영역의 높이가 더 크기 때문에 37:20 상하 분할



- 점심: [(샌드위치, 20)]
- 항목이 한 개 남았으므로 남은 영역에 채워 넣습니다