Data Structure Programming HW5 Report
b08502141 EE2 石旻翰

```
11  def solution(json_input):
12      # --- TODO START --- #
13      json_sum = [0]
14      arr = json_input['array']
15      k = json_input['topk']
16      json_sum.append(arr[0])
17      for i in range(2,len(arr) + 1):
18          json_sum.append(json_sum[i - 1] + arr[i - 1])
19      H = []
20      heapq.heapify(H)
21      for i in range(1, len(arr) + 1):
22          for j in range(i , len(arr) + 1):
23              x = json_sum[j] - json_sum[i - 1]
24              if len(H) < k:
25                  heapq.heappush(H, x)
26              else:
27                  if H[0] < x:
28                      heapq.heappop(H)
29                      heapq.heappush(H, x)
30      # --- TODO END --- #
31      H.sort(reverse=True)
32      # print(H)
33      return H
```

line 17 - 18: To generate json_sum[], it takes O(n).
line 21 - 29: Two-layer for-loop takes O(n^2), and store top-k sums in H takes log(k), so this part takes O(n^2*log(k)).
line 31: len(H) = k, so this sorting takes O(k*log(k)).
To sum up, this program takes O(n^2*log(k)).