

FINALS CYCLES PYTHON

```
def dfs(current, seen, seenSum, best, adj):  
    inPath = False  
    cycleSum = seenSum  
    for i in range(len(seen)-1, -1, -1):  
        if seen[i][0] == current:  
            inPath = True  
            cycleSum -= seen[i][1]  
            break  
  
    if inPath:  
        return cycleSum  
  
    seen.append((current, seenSum))  
  
    for i in adj[current]:  
        best[0] = max(best[0], dfs(i[0], seen, seenSum + i[1], best, adj))  
  
    seen.pop()  
    return best[0]  
  
def find_max_weight(graph):  
    graph = graph.strip().split(' ')  
    newGraph = []  
    for i in range(0, len(graph), 2):  
        newGraph.append(graph[i+0] + graph[i+1])  
    adj = [[] for i in range(26)]  
    nodes = set()  
    for i in newGraph:
```

```
adj[ord(i[0]) - ord('A')].append([ord(i[1]) - ord('A'), int(i[2:])])
nodes.add(ord(i[0]) - ord('A'))
nodes.add(ord(i[1]) - ord('A'))
nodes = list(nodes)
```

[illegible]