

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN
LỚP: CS112.023

Thiết kế thuật toán tìm kiếm các tập con của tập gồm n phần tử

Giảng viên hướng dẫn:

Nguyễn Thanh Sơn

Sinh viên:

Nguyễn Trần Khương An

MSSV:

22520026

Contents

I.	Giới thiệu bài toán.....	3
1.	Xác định yêu cầu	3
2.	Thuật toán BruteForce	3
II.	Ý tưởng	3
III.	Cài đặt	4
IV.	Kiểm thử.....	5
1.	Tạo Test case và đánh giá thời gian.....	5
2.	Đánh giá và kiểm thử.....	6
V.	Nhận xét	6

I. Giới thiệu bài toán

Xây dựng thuật toán theo phương pháp BruteForce để tìm số lượng tập con có tổng các phần tử là số chẵn.

1. Xác định yêu cầu

- **Input:** số nguyên n và mảng A gồm n phần tử
- **Output:** số lượng tập con có tổng là số chẵn
- **Thuật toán:** BruteForce

2. Thuật toán BruteForce

- Brute Force là một thuật toán vét cạn, thuật toán này sẽ chạy tất cả các trường hợp có thể có để giải quyết một vấn đề nào đó (Bao gồm cả trường hợp đúng và các trường hợp sai hay còn gọi là trường hợp dư thừa).
- **Ưu điểm:** đơn giản, dễ hiểu và dễ triển khai
- **Nhược điểm:** độ phức tạp thuật toán lớn và kém hiệu quả

II. Ý tưởng

- Số lượng tập con có thể là 2^n tập con.
- Sử dụng thuật toán BruteForce để vét cạn:
 - + Duyệt qua tất cả tập con có thể là 2^n lần.
 - + Với mỗi lần lặp cần kiểm tra tổng của 1 tập con là chẵn hay lẻ
 - + Tăng biến count lên nếu tập hợp trên thỏa mãn điều kiện
- Đếm count (số lượng tập con thỏa mãn) và in ra

III. Cài đặt

```
import time
import random
import os

def random_array(n):
    return [random.randint(1, 100000) for _ in range(n)]

def sum_even(arr):
    sum = 0
    for i in arr:
        sum += i
    return (sum % 2 == 0)

def generate_subsets(arr):
    subsets = []
    n = len(arr)
    # Duyệt qua tất cả các tập con có thể  $2^n$ 
    for i in range(1, 2**n):
        subset = []
        for j in range(n):
            # Kiểm tra xem phần tử thứ j có thuộc tập con không
            if (i >> j) & 1:
                subset.append(arr[j])
        subsets.append(subset)
    return subsets

def bruteforce(array):
    cnt = 0
    results = []
    n = len(array)
    all_subsets = generate_subsets(array)
    for subset in all_subsets:
        if sum_even(subset):
            cnt += 1
            results.append(subset)
    return cnt, results

def measure_execution_time(algorithm, test_case):
    start_time = time.time()
    cnt, results = algorithm(test_case)
    end_time = time.time()
    return cnt, results, end_time - start_time

n_values = [0, 1, 5, 10, 15, 20, 25, 30]
```

```
for i, n in enumerate(n_values):
    folder_path = "D:\\CS112\\Bài tập 1\\TestCase"
    if not os.path.exists(folder_path):
        os.mkdir(folder_path)
    input_path = os.path.join(folder_path, f"Input{i + 1}.txt" )
    output_path = os.path.join(folder_path, f"Output{i + 1}.txt")
    test_case = random_array(n)
    with open (input_path, "w") as file:
        file.write(str(n) + "\n")
        file.write(" ".join(map(str, test_case)) + "\n")
        file.close()
    # print(test_case)
    cnt, results, bruteforce_time = measure_execution_time(bruteforce, test_case)
    with open (output_path, "w") as file:
        file.write(f"Count: {cnt}" + "\n")
        file.write(f"Time: {bruteforce_time}" + "\n")
        file.close()
```

IV. Kiểm thử

1. Tạo Test case và đánh giá thời gian

- Tạo các test case lần lượt có $n = [0, 1, 5, 10, 15, 20, 25, 30]$.
- Chi tiết bộ test trong folder Testcase.
- Đo thời gian chạy ứng với từng giá trị n :

Number of element	0	1	5	10	15	20	25	30(1)
Time	0.00	0.00	0.00	0.002	0.059	3.139	124.408	$+\infty$

*Lưu ý: (1) Không thể kiểm tra vì cấu hình máy và thời gian thực hiện quá tải.

```
Traceback (most recent call last):
  File "d:\CS112\Bài tập 1\brute_force.py", line 58, in <module>
    cnt, results, bruteforce_time = measure_execution_time(bruteforce, test_case)
    ~~~~~^~~~~~
  File "d:\CS112\Bài tập 1\brute_force.py", line 40, in measure_execution_time
    cnt, results = algorithm(test_case)
    ~~~~~^~~~~~
  File "d:\CS112\Bài tập 1\brute_force.py", line 31, in bruteforce
    all_subsets = generate_subsets(array)
    ~~~~~^~~~~~
  File "d:\CS112\Bài tập 1\brute_force.py", line 24, in generate_subsets
    subsets.append(subset)
MemoryError
```

2. Đánh giá và kiểm thử

- Đánh giá độ phức tạp thuật toán:
 - + Vòng lặp thứ nhất duyệt từ 1 tới 2^n nên độ phức tạp là $O(2^n)$
 - + Vòng lặp thứ 2 duyệt từ 1 tới n nên độ phức tạp là $O(n)$.

Vậy độ phức tạp thuật toán trong mọi trường hợp là $O(n \cdot 2^n)$.

V. Nhận xét

Thuật toán Brute Force được sử dụng để giải quyết bài toán tìm tất cả các tập con có tổng chẵn từ một dãy số cho trước. Mặc dù thuật toán này dễ hiểu và dễ triển khai, nhưng đối với dữ liệu lớn, nó có độ phức tạp thời gian cao. Việc kiểm tra tất cả các tập con có thể tạo ra mất nhiều thời gian, đặc biệt là với số lượng phần tử lớn. Điều này làm cho thuật toán không hiệu quả trong một số trường hợp.