Bit Masks
(Programming Club 3)

Tomasz Kosciuszko

04.02.2019

1 Introduction

Life is tough - this truth you often learn when solving problems in programming and not only. Sometimes the only option remaining is brute force: iterate over all possible solutions and find the one working.

Consider the following question, how to iterate over all subsets of a set with n elements. For example for $n = 3$:

- {}
- {1}
- {2}
- {3}
- {1,2}
- {1,3}
- {2,3}
- {1,2,3}

Let us represent each number above as either 1 if it appears in the subset or 0 otherwise. Then {} translates to 000 and {1,3} to 101 and so on... These strings of 0s and 1s are just number in the binary representation. So we can iterate over all numbers from 0 to $7 = 2^3 - 1$ and print sets that correspond to each of them.

2 Example problem

Imagine you work for package delivery. You can transport up to 200kg in your car and the parcels that you have to deliver today have weights: 13, 76, 40, 82, 19, 67, 26, 39, 50. What is the heaviest set of parcels that you can take without exceeding the limit?

3 Solution

Let's not try anything too sophisticated, we will just iterate over all possible subsets and check whether their sum fits into the car. We start with the function *get_subset*(n, k) which will give us the list of k 0s and 1s corresponding to n:

```
def get_subset(n, k):
    power_of_2 = 1
    result = []
    for i in range(k):
        if (n & power_of_2) > 0:
            result.append(1)
        else:
            result.append(0)
        power_of_2 = power_of_2 * 2
    return result
```

Now let's create a list with available packages and iterate through all subsets (there are 2^9 of them):

```
parcels = [13, 76, 40, 82, 19, 67, 26, 39, 50]
limit = 200
result = -1  # The best result so far.
result_list = []
for i in range(2**len(parcels)):
    mask = get_subset(i, len(parcels))
    weight = sum([mask[j] * parcels[j]
        for j in range(len(parcels))])
    if weight <= limit and weight > result:
        result = weight
        result_list = [] # Clear the list.
        for j in range(len(parcels)):
            if mask[j] == 1:
                result_list.append(parcels[j])

print ("You can take at most %s kg" % (result,))
print ("Pack the following: %s"  % (result_list,))
```

4 Further problems

Keep in mind that this is just a simple example of how to represent states as binary numbers. Bit Masks can be used in much more difficult problems, for example the last one from below:

- Easy: `http://codeforces.com/problemset/problem/579/A`
- Medium: `http://codeforces.com/problemset/problem/114/B`
- Medium: `https://www.hackerrank.com/challenges/sum-vs-xor/problem`
- Hard: `http://codeforces.com/problemset/problem/678/E`