

برای حل این مسئله با روش جستجوی کامل باید تمام زیرمجموعه‌های مجموعه عناصر را تولید کنیم و شرط مسئله را در هریک از آن‌ها بررسی کنیم و بهترین حالت را پیدا کنیم. اگر تولید هر یک از زیرمجموعه‌های این مجموعه در زمان $O(1)$ اتفاق بیفتد و بررسی شرط مسئله در زمان ثابت انجام شود، کارایی زمانی الگوریتم $\Theta(2^n)$ خواهد بود. در زیر روشی برای تولید زیرمجموعه‌های یک مجموعه آورده شده است:

Listing 1: Python Powerset Implementation

```
def powerset(elements: list):  
    p = [[]]  
    for element in elements:  
        for i in range(len(p)):  
            p.append(p[i] + [element])  
    return p
```

بدین صورت که ابتدا با مجموعه که تنها مجموعه تهی را دارد شروع کرده و عضو اول مجموعه را به آن اضافه کرده و به زیرمجموعه‌های مجموعه این تک عضوی دست می‌یابیم، حال با اضافه کردن عضو بعدی به این مجموعه‌ها و اضافه کردن آن‌ها به لیست مجموعه‌ها، لیست زیرمجموعه‌های این دو عضو را داریم و با ادامه این روند، تمام زیرمجموعه‌های این مجموعه تولید خواهند شد.