

노트북: 첫 번째 노트북
만든 날짜: 2020-12-01 오전 3:44
URL: <https://continuous-development.tistory.com/204>

Algorithm

[Algorithm] 38강 : 에라토스테네스의 체 알고리즘의 정의와 구현

2020. 12. 1. 02:17 수정 삭제 공개

에라토스테네스의 체 알고리즘

1.1 에라토스테네스의 체 알고리즘이란?

- 다수의 자연수에 대하여 소수 여부를 판별할 때 사용하는 대표적인 알고리즘이다.
- N보다 작거나 같은 모든 소수를 찾을 때 사용할 수 있다.

1.2 에라토스테네스의 체 동작 알고리즘

- 동작과정
 - 2부터 N까지의 모든 자연수를 나열
 - 남은 수 중에서 아직 처리하지 않은 가장 작은 수 I를 찾는다
 - 남은 수 중에서 I의 배수를 모두 제거한다
 - 더 이상 반복할 수 없을때까지 2번과 3번의 과정을 반복한다.

Step 1] 아직 처리하지 않은 가장 작은 수 2를 제외한 2의 배수는 모두 제거합니

2	3	4	5	6
7	8	9	10	11
12	13	14	15	16
17	18	19	20	21
22	23	24	25	26

[Step 2] 아직 처리하지 않은 가장 작은 수 3을 제외한 3의 배수는 모두 제거합니

2	3	4	5	6
7	8	9	10	11
12	13	14	15	16
17	18	19	20	21
22	23	24	25	26

Step 4] 마찬가지로의 과정을 반복했을 때 최종적인 결과는 다음과 같습니다.

2	3	4	5	6
7	8	9	10	11
12	13	14	15	16
17	18	19	20	21
22	23	24	25	26

1.3 알고리즘 구현

```
import math

n = 1000

array = [True for i in range(n + 1)]

for i in range(2, int(math.sqrt(n) + 1)):
    if array[i] == True:
        j = 2
        while i * j <= n:
            array[i * j] = False
            j += 1

for i in range(2, n + 1):
    if array[i]:
        print(i, end=' ')
```

1.4 알고리즘 성능 분석

- 에라토스테네스의 체 알고리즘의 시간 복잡도는 사실상 선형 시간에 가까운 정도로 매우 빠름
 - 시간 복잡도는 $O(N \log \log N)$ 이다.
- 에라토스테네스의 체 알고리즘은 다수의 소수를 찾아야 하는 문제에서 효과적
 - 하지만 각 자연수에 대한 소수 여부를 저장해야 하므로 메모리가 많이 필요

이 자료는 동빈 나 님의 **이코 테** 유튜브 영상을 보고 정리한 자료입니다.

참고 : www.youtube.com/watch?v=m-9pAwq1o3w&list=PLRx0vPvIEmdAghTr5mXQxGpHjWqSz0dgC

'Algorithm' 카테고리의 다른 글

[Algorithm] 38강 : 에라토스테네스의 체 알고리즘의 정의와 구현

[Algorithm] 37강 : 소수 판별 알고리즘의 정의와 구현

[Algorithm] 36강 : 위상정렬 알고리즘의 정의와 구현

[Algorithm] 35강 : 크루스칼 알고리즘의 정의와 구현

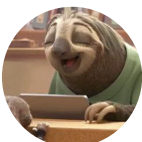
[Algorithm] 34강 : 서로소 집합을 활용한 사이클 판별

[Algorithm] 33강 : 서로소 집합 자료구조의 정의와 구현

에라토스테네스의 체

에라토스테네스의 체 알고리즘

에라토스테네스의 체 알고리즘 구현



나아무늘보

혼자 끄적끄적하는 블로그 입니다.

