

## روش اول

از `schedule(static)` برای تقسیم اجراهای حلقه ها استفاده شده است اما با توجه به اینکه روند اجرا نخ ها نا مشخص است در حین اجرا چک می کنیم که نخ دوم حتما در ابتدای کار مقدار پایانی متغیر قبلی را به عنوان ورودی در نظر نگیرد و به جای آن صرفا کاری انجام ندهد. برای پیاده سازی از ۲ نخ استفاده شده است. و در این حالت موازی پس از پایان کار به صورت سریال تمامی اعضای نیمه دوم آرایه را با عضو انتهایی نیمه اول (که مجموع تمام اعضای آن است) جمع می کنیم. تا مقادیر درستی در خروجی نهایی داشته باشیم. اگر تعداد نخ ها ۴ تا بود نیز باید ابتدا مانند همین حالت تکرار ها را تقسیم می کردیم و در انتهای کار مقدار جمع تجمعی هر بخش را به تمام اعضای بخش بعدی آن اضافه می کردیم تا زمانی که به انتهای آرایه برسیم.

زمان اجرای حالت سریال به ازای ورودی یک میلیارد به صورت زیر می باشد.

```
[-] Please enter N:1000000000
Time Elapsed: 32.945000 Secs
```

زمان اجرای حالت موازی به ازای ورودی یک میلیارد به صورت زیر می باشد.

```
[-] Please enter N:1000000000
Time Elapsed: 19.058000 Secs
```

همان طور که مشخص است زمان اجرا در حالت موازی حدودا نصف زمان اجرا در حالت سریال میباشد.

## روش دوم

برای پیاده سازی این روش هر حلقه **for** درونی را یک ناحیه موازی در نظر گرفته و آن را با یک نخ پردازش میکنیم سپس زمانی که به تسک رسیدیم آن را ساخته و کار را ادامه می دهیم. سپس در انتهای هر حلقه **for** درونی که پایان هر بار اجرا به ازای هر طول گام است صبر می کنیم تا تمامی **task** ها به پایان برسند سپس کار را ادامه می دهیم. چون برخی خانه های آرایه در هر بار اجرا بیش از یکبار فراخوانی می شوند. برای ایجاد هماهنگی در دور اول از **a** می خوانیم و در **b** میریزیم، در دور دوم از **b** می خوانیم در **a** میریزیم و به همین صورت تا انتها ادامه می دهیم. و در انتهای کار آرایه ای که نتایج نهایی در آن وجود دارند را چاپ می کنیم. برای پیاده سازی این بخش از ۲ نخ استفاده شده است.

زمان اجرای حالت سریال به ازای ورودی یک میلیارد به صورت زیر می باشد.

```
[ - ] Please enter N:1000000000  
Time Elapsed: 120.513000 Secs
```

زمان اجرای حالت موازی به ازای ورودی ۱۰۰ میلیون به صورت زیر می باشد. (پس از ۲ ساعت اجرای ۱ میلیارد نتیجه ای بدست نیامد)

```
[ - ] Please enter N:1000000000  
Time Elapsed: 1461.347000 Secs
```

با توجه به اینکه عملکرد الگوریتم دوم کاملاً وابسته به تعداد نخ های موجود در سیستم است. و اینکه پردازنده تعداد نخ های محدودی دارد. و همچنین با توجه به اینکه خود این الگوریتم کند تر از الگوریتم قبلی است باعث می شود که زمان اجرا آن بیشتر از حالت قبلی باشد. مشخصاً اگر در **gpu** این برنامه را اجرا کنیم با سرعت بالایی تمام تسک ها اجرا شده و می توانیم به دور بعدی برویم در حالی که اگر الگوریتم قبلی در **gpu** انجام شود باید زمان زیادی را در انتهای هر کار منتظر بمانیم تا کار جمع های نهایی که باید به صورت سریال انجام شود به پایان برسد.