

## زمان‌های اجرا:

الگوریتم \ n	سریال	موازی سازی شده	Hillis and Steele
1024	0.000017	0.000372	0.000101
65536	0.000884	0.001268	0.007341
1048576	0.010212	0.015195	0.045577
33554432	0.248628	0.185826	2.015626

## تحلیل:

همانطور که مشاهده می‌کنید، الگوریتم موازی شده، تا حدود ۱.۳۴ افزایش سرعت پیدا می‌کند، دلیل کم بودن این ضریب اول کم بودن تعداد عملیات‌های روش سریال و کم بودن تعداد thread ها (۴) و دوم موازی نشدن بخش پر کردن آرایه می‌باشد.

همچنین روش Hillis and Steele افزایش سرعت نداشته زیرا تحلیل زمانی این الگوریتم

$O(n \times \log_2(n))$  می‌باشد و چون تعداد thread ها کمتر از  $\log_2(n)$  می‌باشد پس این

الگوریتم باعث افزایش سرعت نمی‌شود، این الگوریتم در حالتی که تعداد thread ها بیشتر از

$\log_2(n)$  باشد باعث speed up می‌شود.

به دلیل اینکه این الگوریتم در حلقه درونی‌اش تعداد زیادی جمع انجام می‌دهد که این جمع‌ها می‌توانند برداری انجام شوند (جمع یک آرایه با شیفت پیدا کرده همان آرایه) این سری جمع‌ها می‌توانند به صورت

تک دستور در GPU انجام شوند. این الگوریتم می‌تواند با پیچیدگی زمانی  $O(c \times \log_2(n))$

محاسبات را انجام دهد که  $C$  تعداد دستورات لازم برای انجام حلقه‌ی دوم توسط GPU می‌باشد. اگر

امیرحسین پاشایی هیر ۹۷۳۱۰۱۳ - محمد مهدی منتظر ۹۷۳۱۱۲۰

$C$  عدد کمی باشد (برای مثال اگر GPU بتواند ۱۰۰۰ عملیات جمع انجام دهد در هر دستور و  $n$  حدود ۱ میلیون باشد مقدار  $C = 1000$  می شود) این الگوریتم سرعت را به خوبی افزایش می دهد.