

Program Maximum3

```
    assign  $\langle \mid j: 0 \leq j < M:: A[j] = \max(A[2j], A[2j+1]) \rangle$   
end {Maximum3}
```

- Čo vieme povedať o prvkoch poľa (okrem $A[0]$) v
- čase $O(\log N)$?

- Sformulujte nejakú safety podmienku a ukáže že platí
- Sformulujte nejakú progress podmienku a ukáže že platí
- Ako bude vyzeráť pole keď výpočet dosiahne FP

Program Maximum4

```
    assign  $\langle \mid j: 0 \leq j < M:: A[j] = \max(A[2j], A[2j+1]) \rangle$   
End
```

Bude program fungovať? Prečo?

Program Reach

declare r : array[V] of boolean

initially $\langle \forall v: v \in V :: r[v] = (v = \text{init}) \rangle$

assign $\langle \forall u, v: (u, v) \in E :: r[v] := r[u] \vee r[v] \rangle$

end{Reach}

Pomocou safety a progres vlastností dokážte, že program dosiahne FP a že to vtedy vypočíta, čo bola úloha.

Bude program P triediť pole? Ak nie, prečo, ak áno, odôvodnite, v akej zložitosti ak by sme zaručili striedanie príkazov a s koľkými procesormi?

Program P

assign

$\langle \parallel i: 1 \leq i < N-3 \wedge \text{even}(i) :: y[i], y[i+1], y[i+2], y[i+3] := \text{sort4}(y[i], y[i+1], y[i+2], y[i+3]) \rangle$

$\parallel \langle \parallel i: 1 \leq i < N-3 \wedge \text{odd}(i) :: y[i], y[i+1], y[i+2], y[i+3] := \text{sort4}(y[i], y[i+1], y[i+2], y[i+3]) \rangle$

end{P4}

Upravte Rank Sort, tak aby nemal always sekciu ale triedenie bolo realizované v časti assign.

Upravte program Rank Sort. tak, aby vedel triediť aj polia, kde nemusia byť všetky prvky rôzne. Odôvodnite správnosť riešenia.