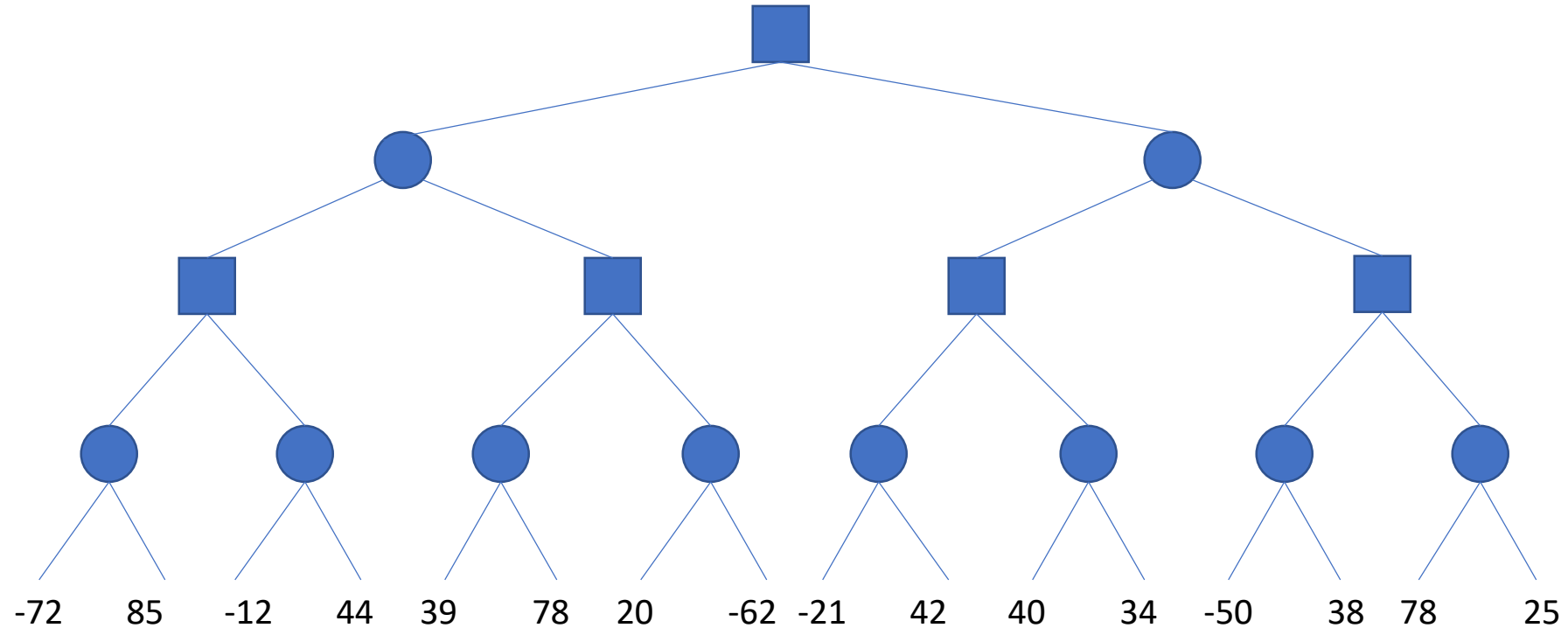
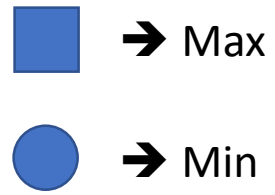


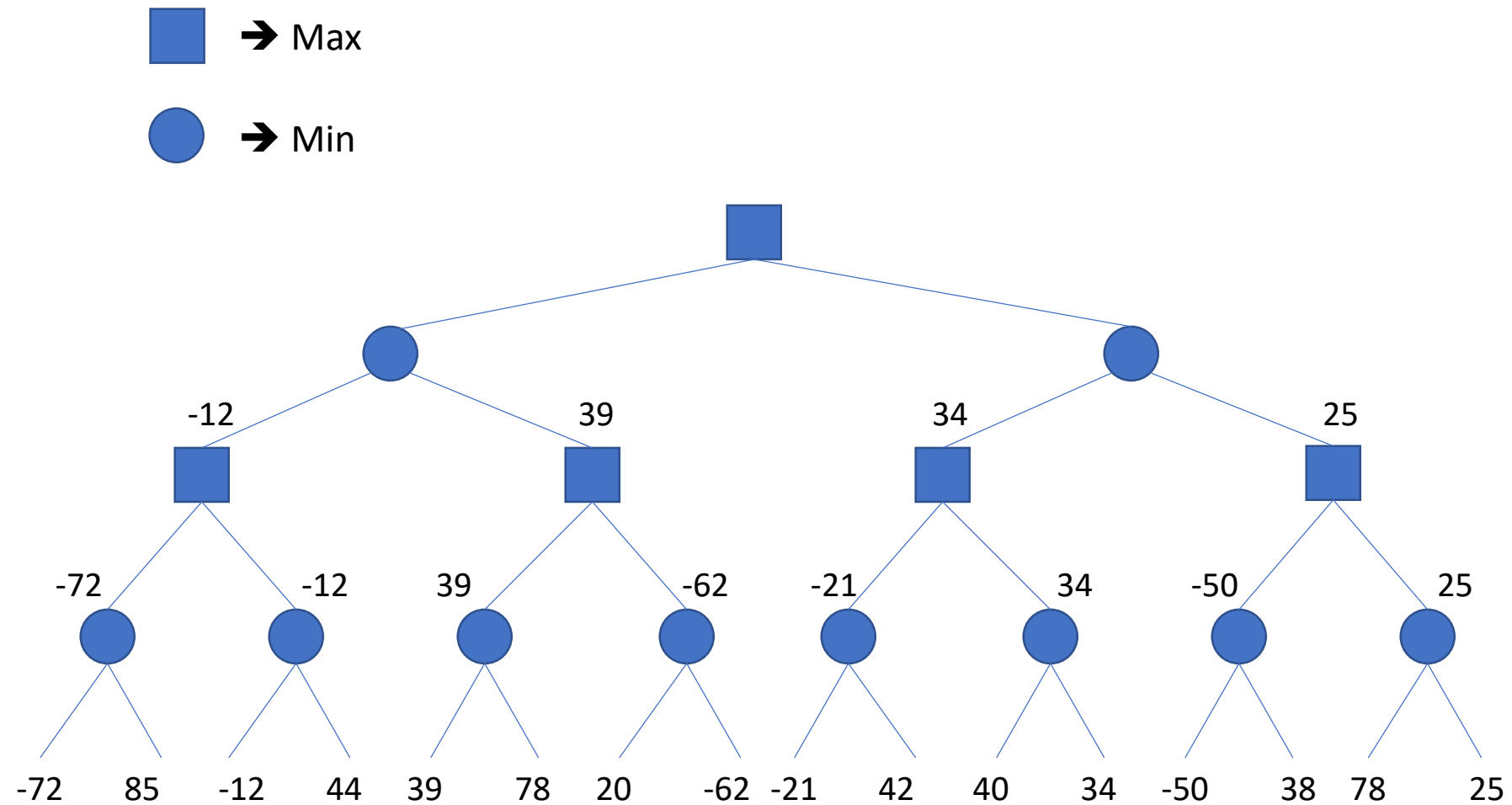
Minimax Algorithm



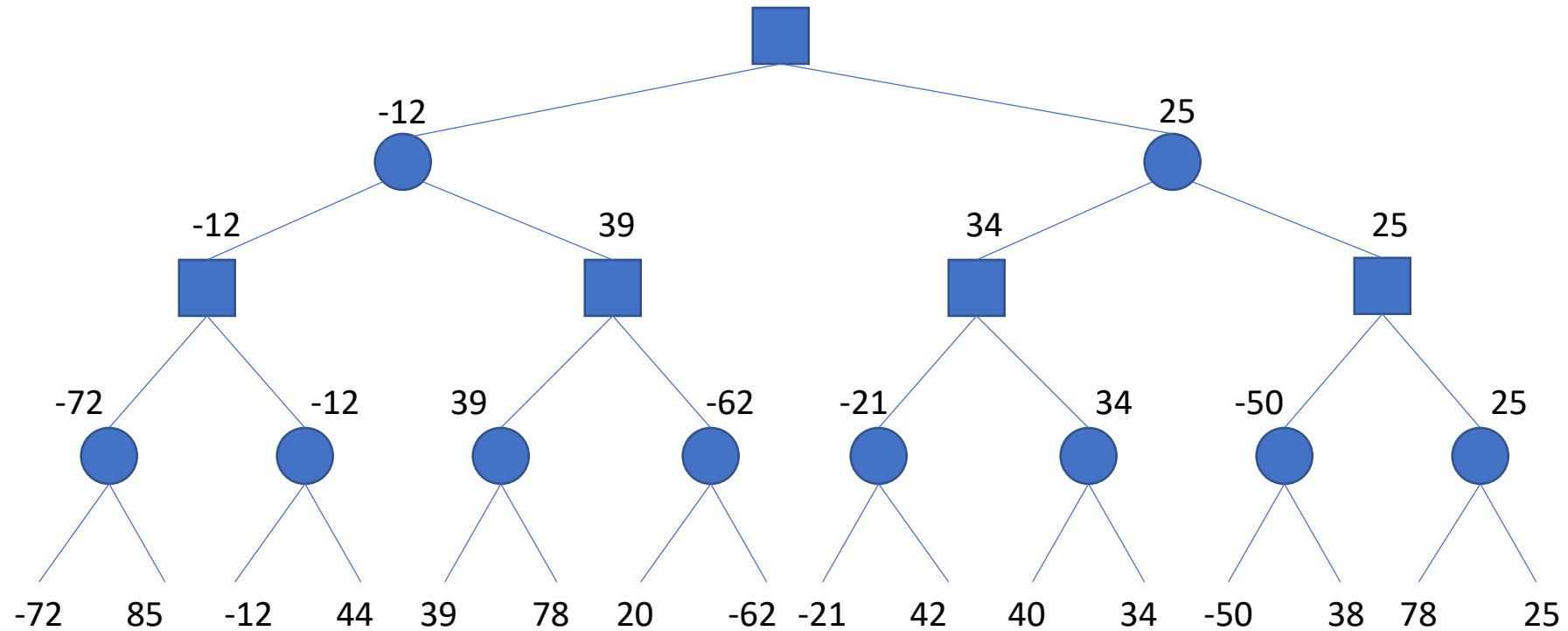
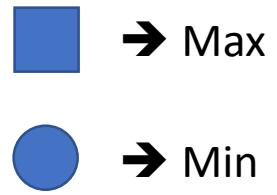
Solve the above tree using Minimax Algorithm with and without Alpha Beta Pruning!

Minimax algorithm:

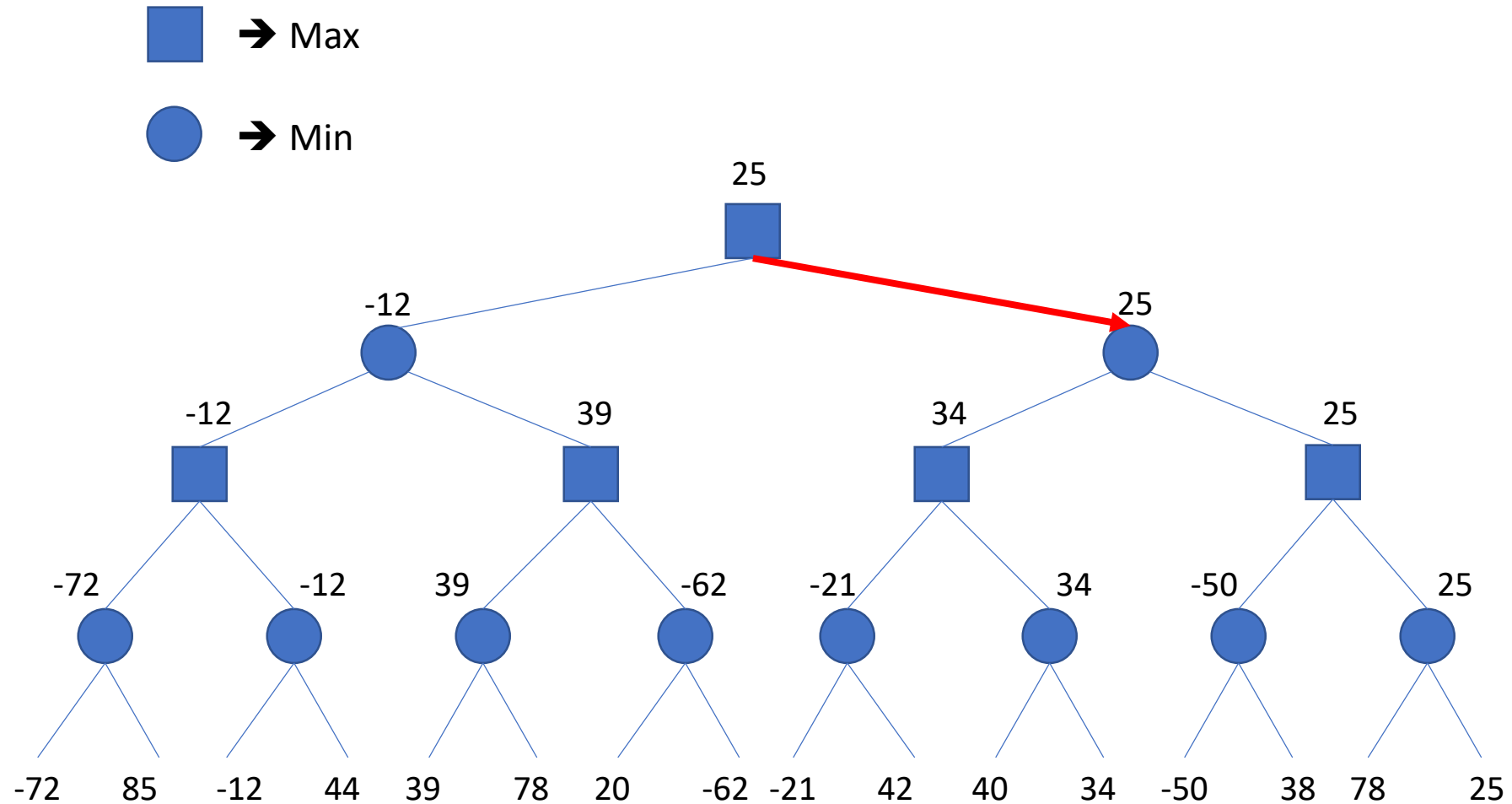
- Untuk setiap Min, Pilih angka yang paling kecil dari anak anaknya
- Untuk setiap Max, Pilih angka yang paling besar dari anak anaknya



- Kemudian hitung angka maximum, di level Max

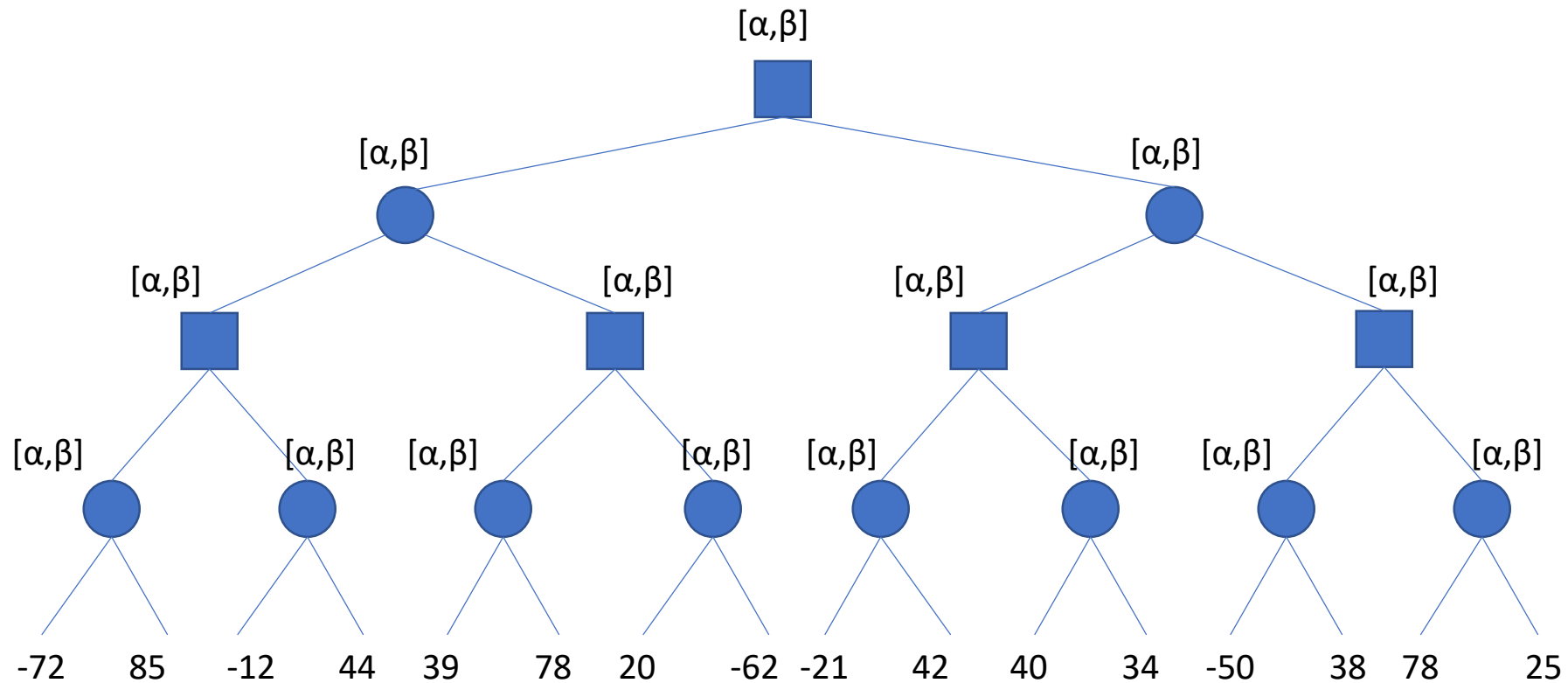


- Kemudian Hitung angka minimumnya lagi



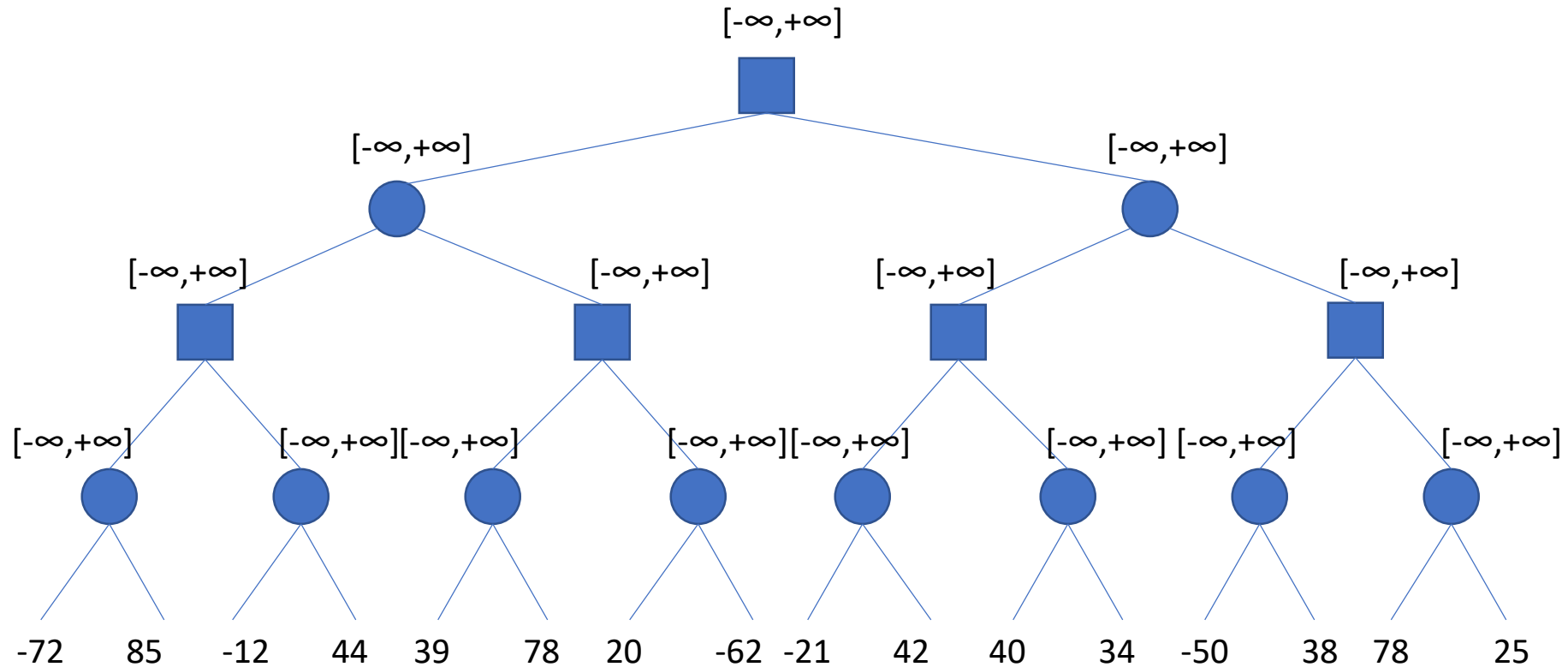
- Untuk max yang terakhir, pilih node yang memberikan nilai terbesar. Node tersebut akan menjadi pilihan jalan yang selanjutnya

Alpha Beta Pruning

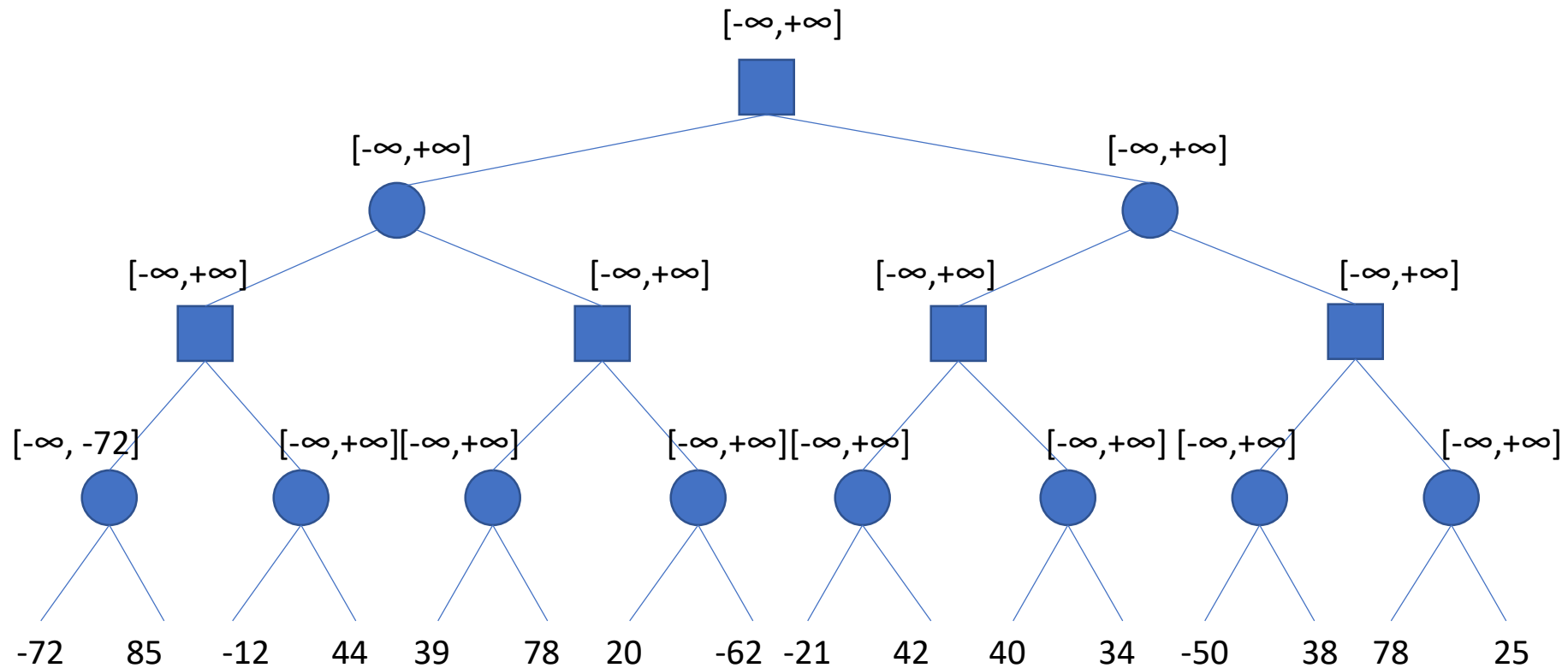


Alpha Beta Pruning

- Merupakan algoritma untuk menambah efisiensi dari algoritma minimax
- Setiap node mempunyai dua tambahan variabel: α, β
- Dua variabel tersebut berguna sebagai variabel pembanding untuk menentukan apakah cabang dari sebuah pohon akan dipotong (pruning) atau tidak

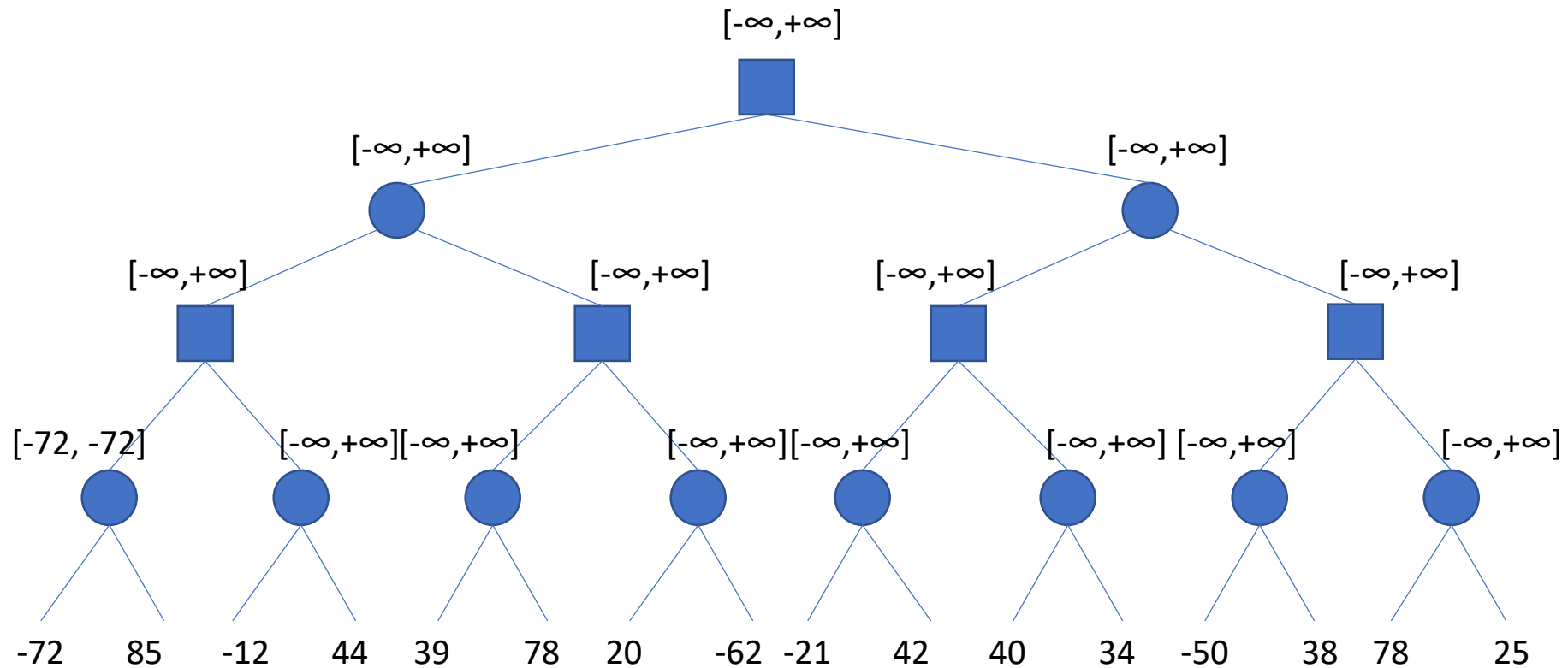


- α dipakai untuk mencari nilai terbesar
- β dipakai untuk mencari nilai terkecil
- Oleh karena itu, α diinisiasi dengan nilai $-\infty$ dan β diinisiasi dengan nilai $+\infty$



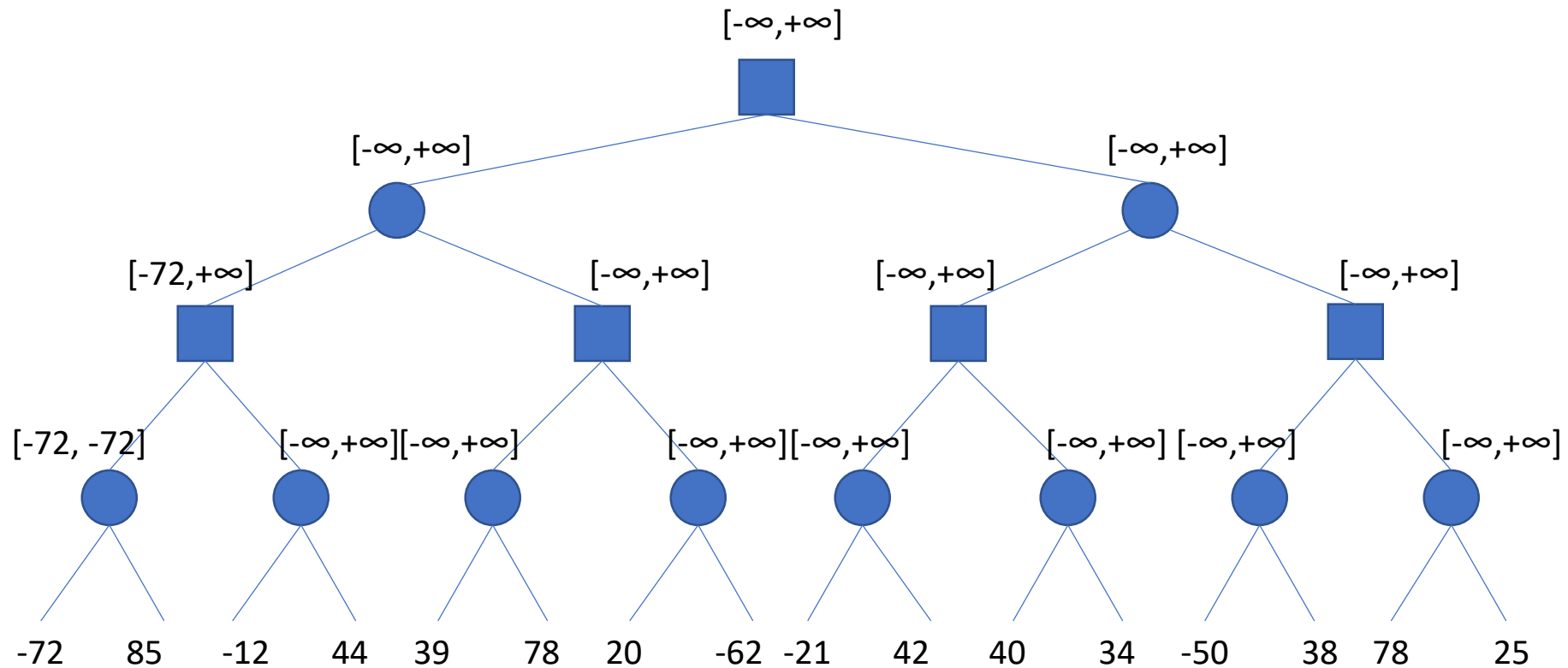
Untuk node Min:

- Bandingkan β dengan nilai dari setiap children node dari node Min
- Apabila nilai β lebih besar dari nilai children node, ubah β dengan nilai children node
 - $\beta = +\infty \rightarrow +\infty > -72 \rightarrow \beta = -72$



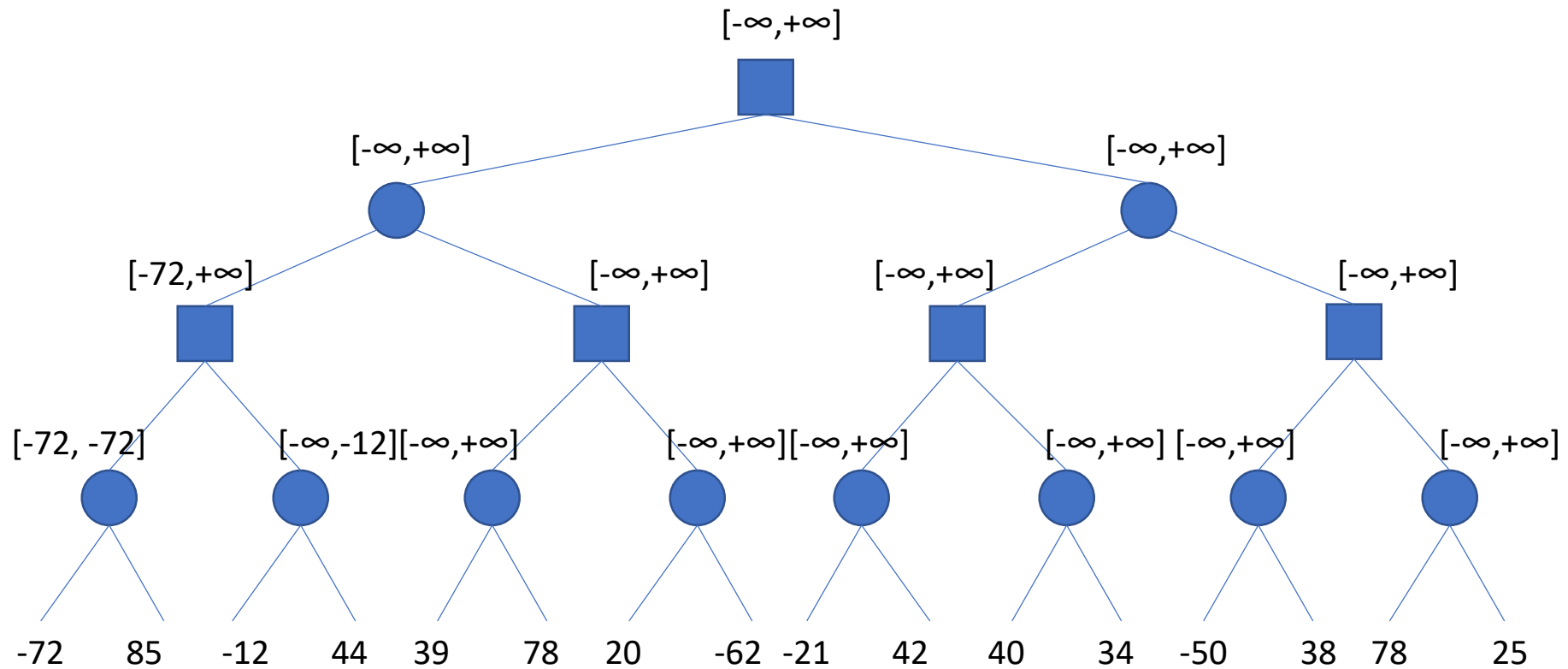
Untuk node Min:

- Bila semua children node sudah dibandingkan dengan nilai β , pindahkan nilai β ke α di node min
- Nilai α di Min akan menjadi nilai dari node tersebut

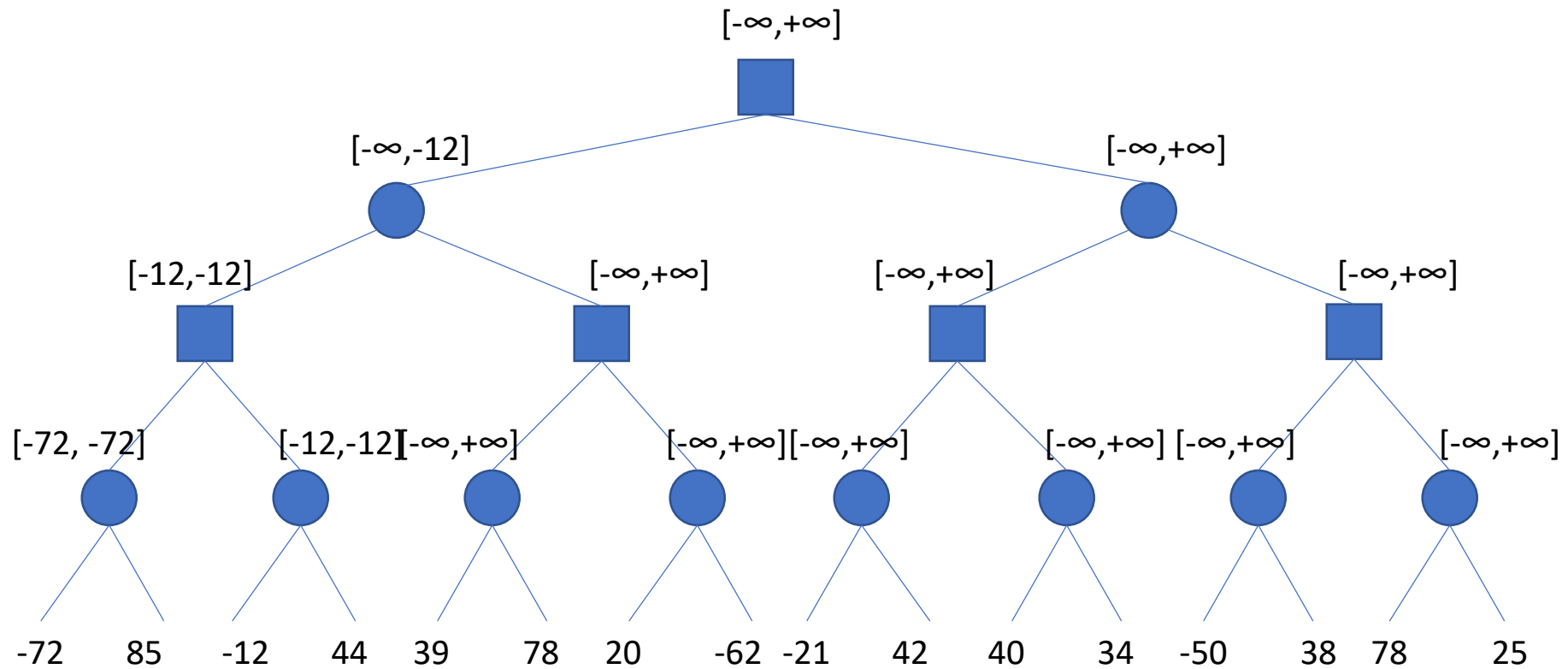


Untuk node Max:

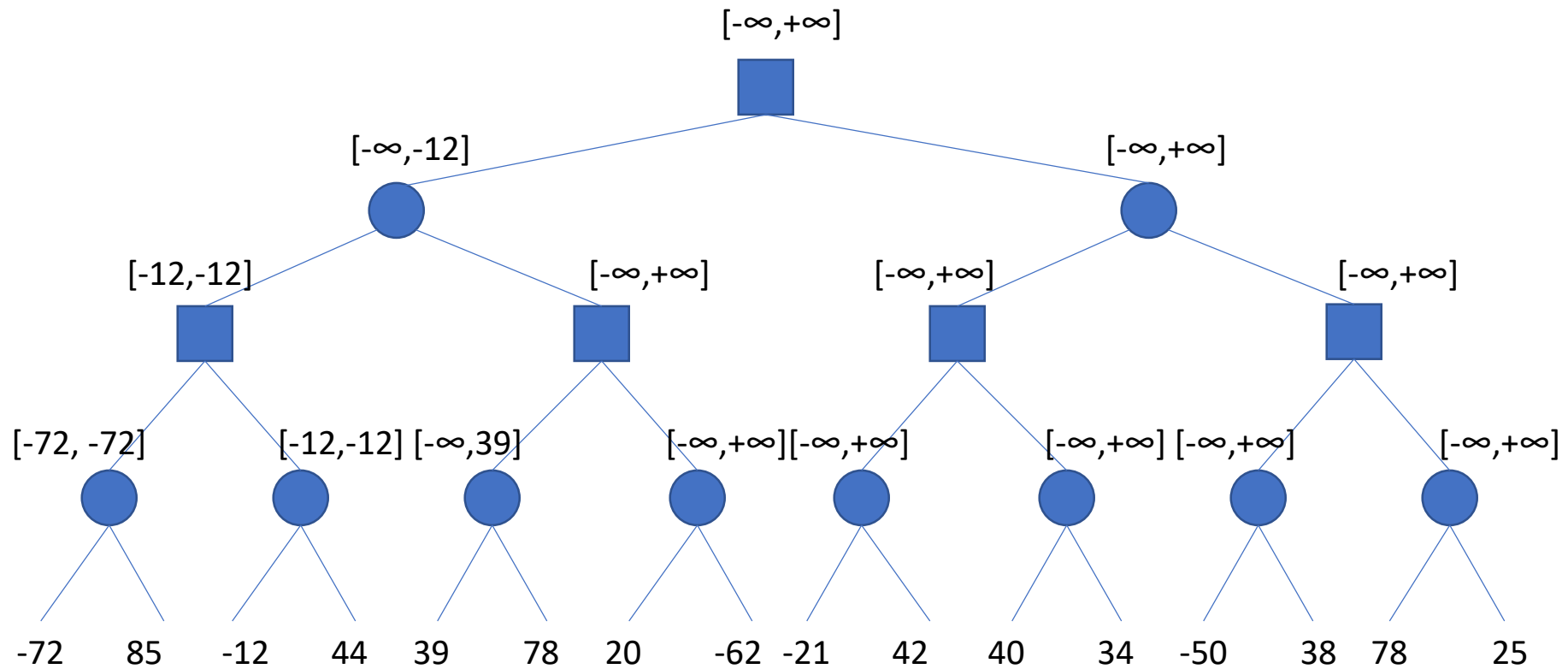
- Bandingkan α dengan nilai dari setiap children node dari node Max
- Apabila nilai α lebih besar dari nilai children node, ubah α dengan nilai children node
 - $\alpha = -\infty \rightarrow -\infty < -72 \rightarrow \alpha = -72$



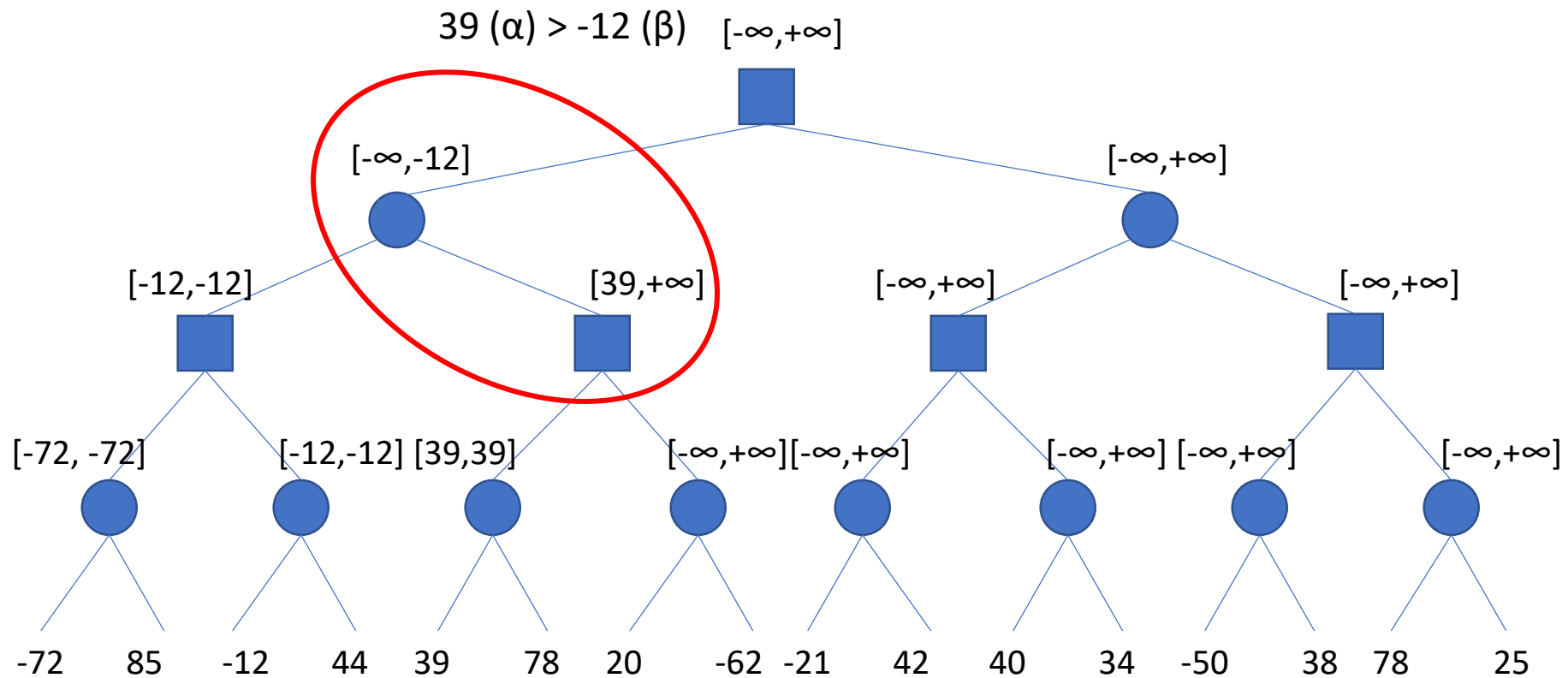
Lakukan hal yang sama seperti sebelumnya sampai ...



Lakukan hal yang sama seperti sebelumnya sampai ...

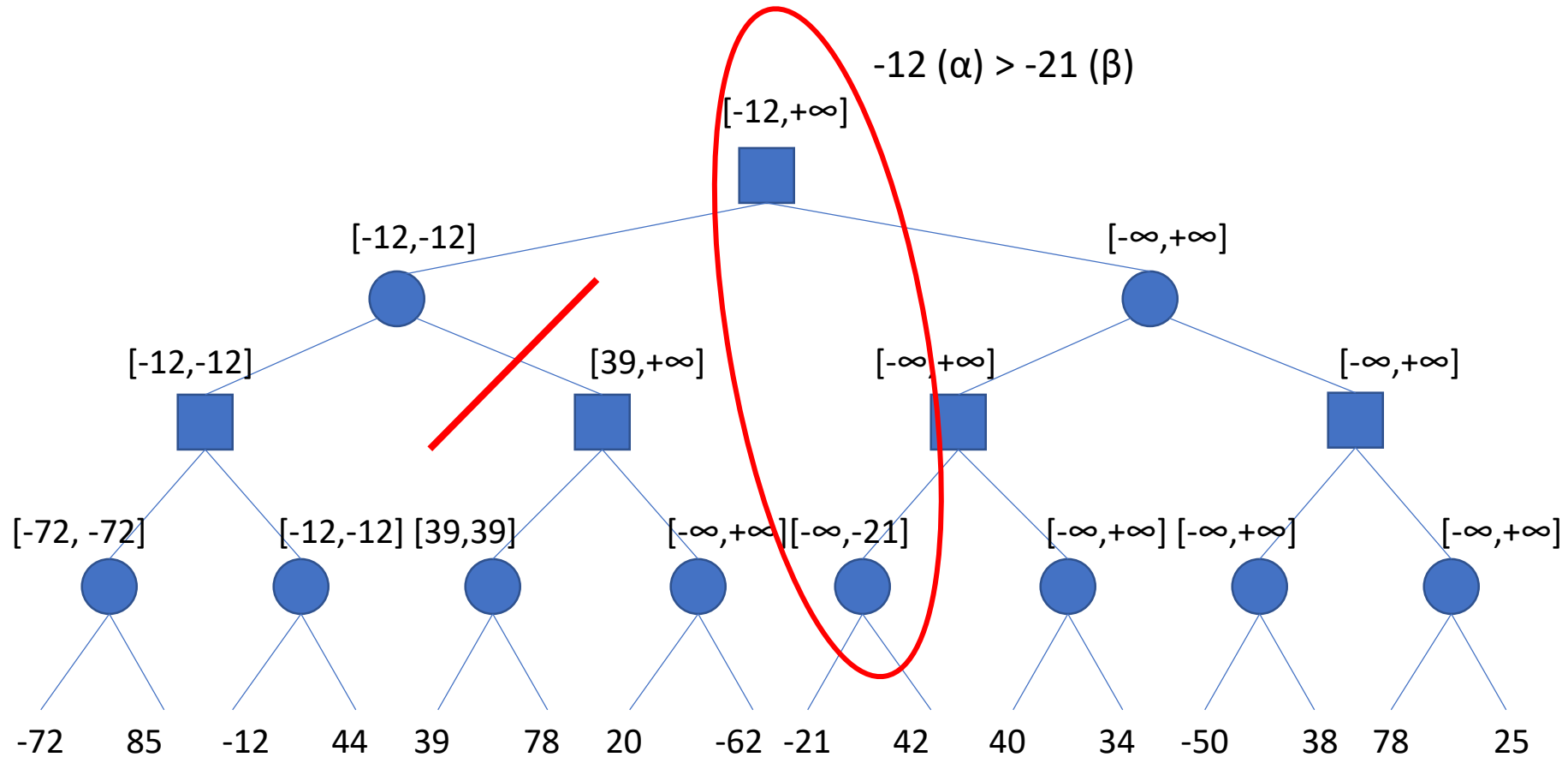


Lakukan hal yang sama seperti sebelumnya sampai ...



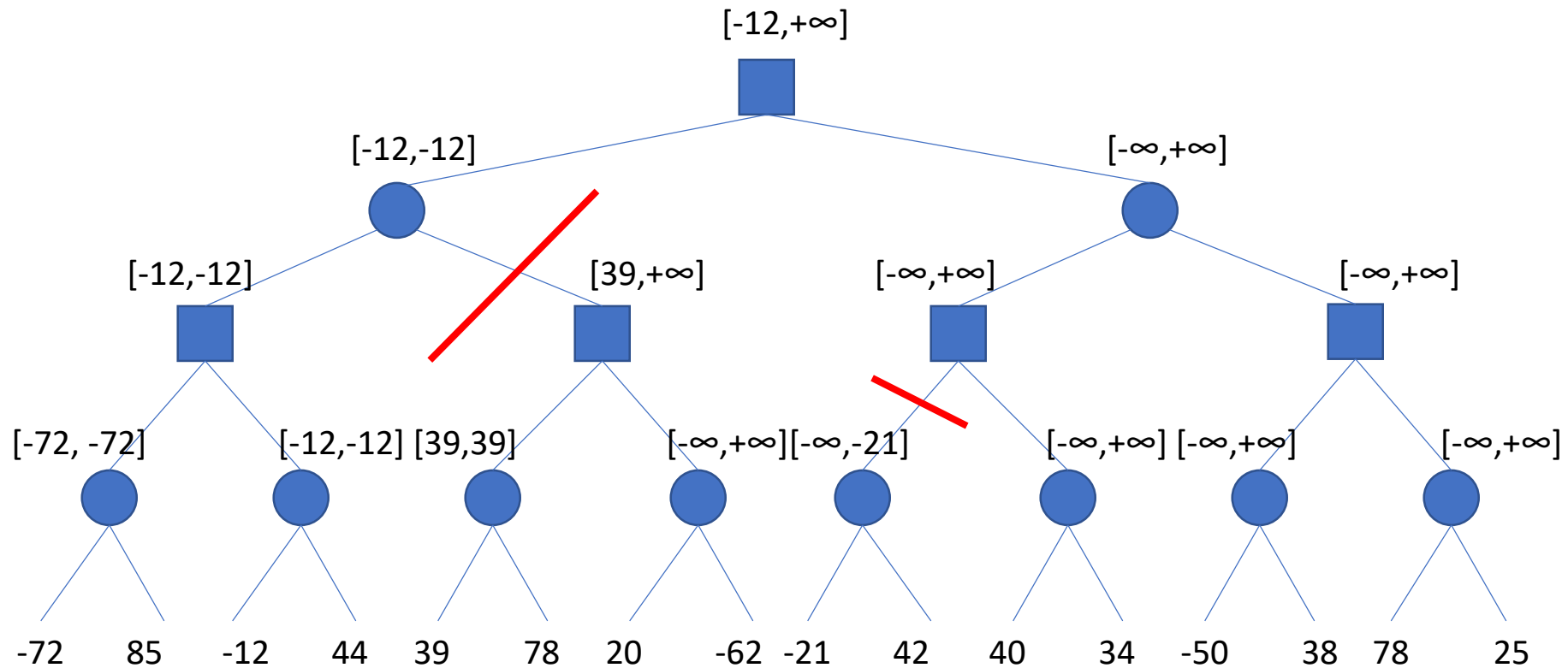
Lakukan hal yang sama seperti sebelumnya sampai kita menemukan:

- Posisi di max $\rightarrow \alpha$ di Max lebih besar dari β di Min di atasnya atau
- Posisi di min $\rightarrow \beta$ di Min lebih kecil dari α di Max yang di atasnya



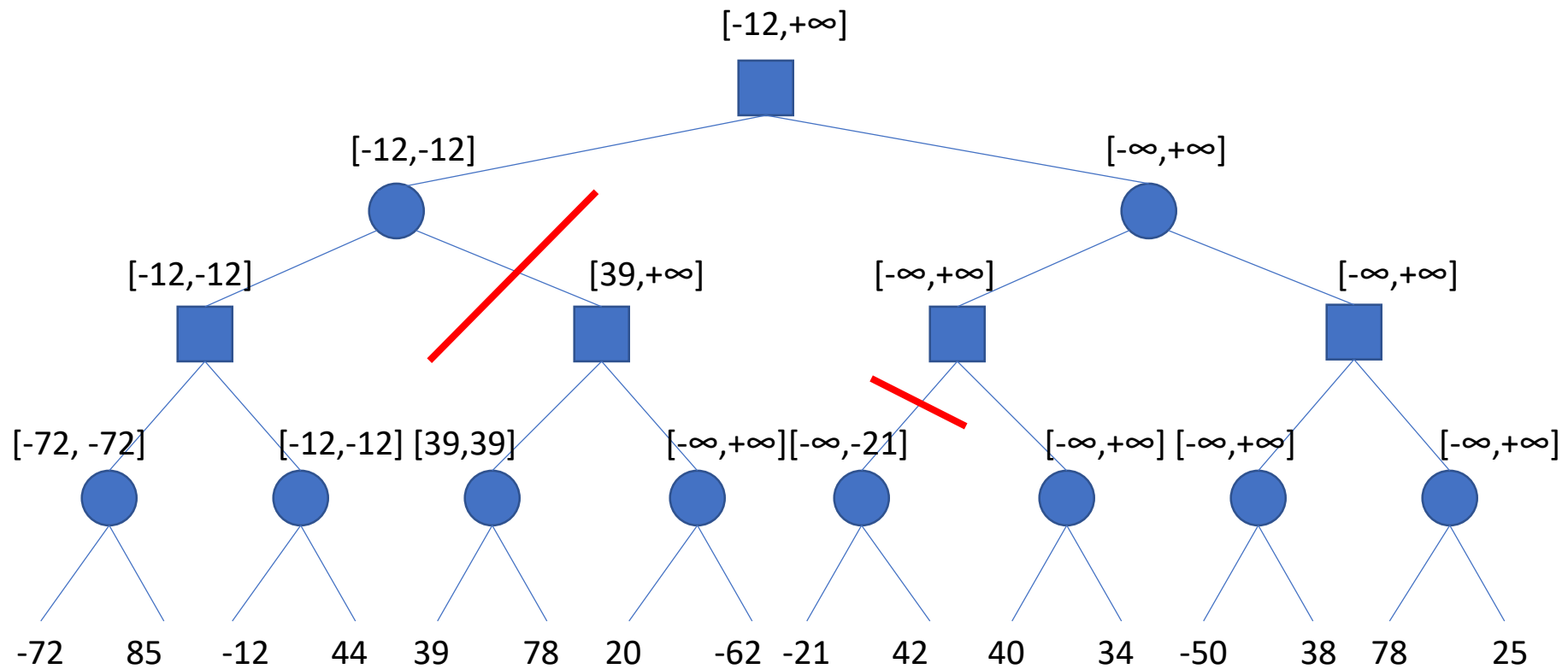
Lakukan hal yang sama seperti sebelumnya sampai kita menemukan:

- Posisi di max $\rightarrow \alpha$ di Max lebih besar dari β di Min di atasnya atau
- Posisi di min $\rightarrow \beta$ di Min lebih kecil dari α di Max yang di atasnya

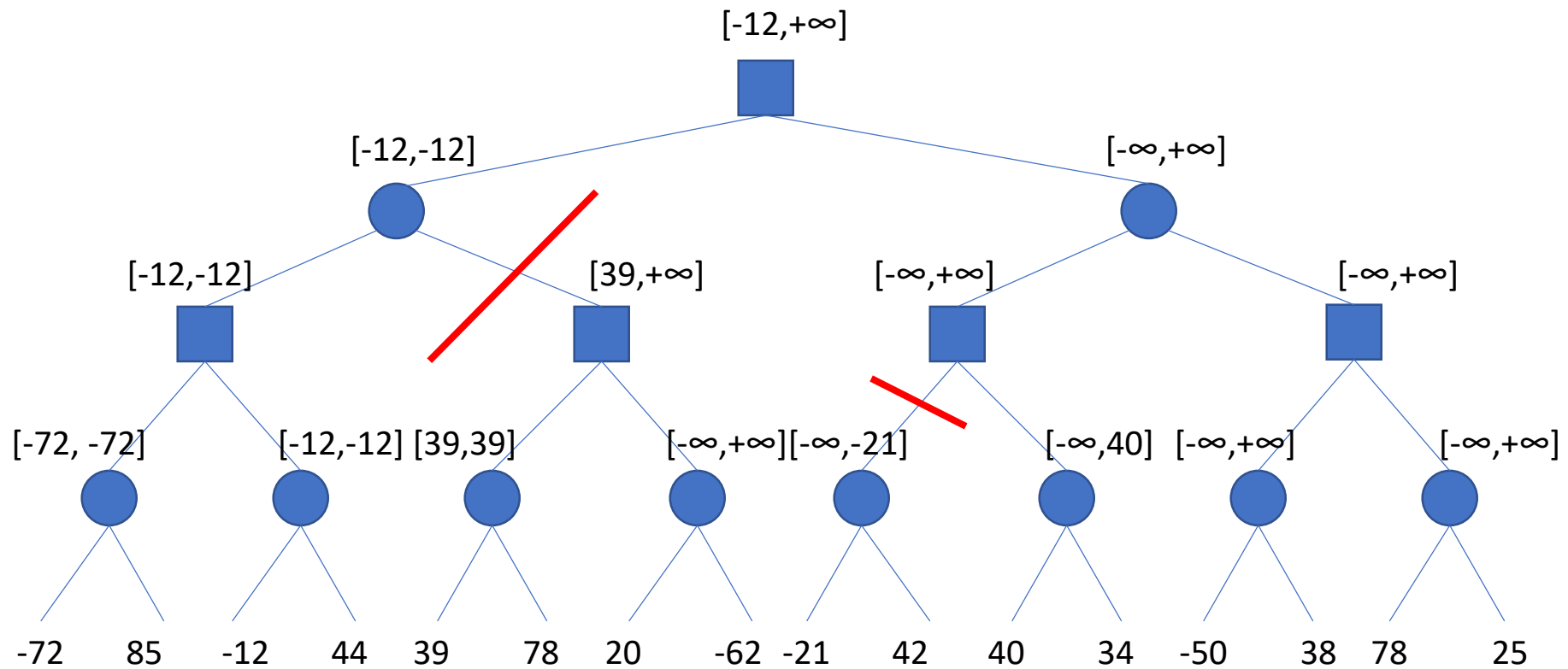


Apabila ditemukan maka potong cabang pohon yang menyambungkan Max dan Min tersebut

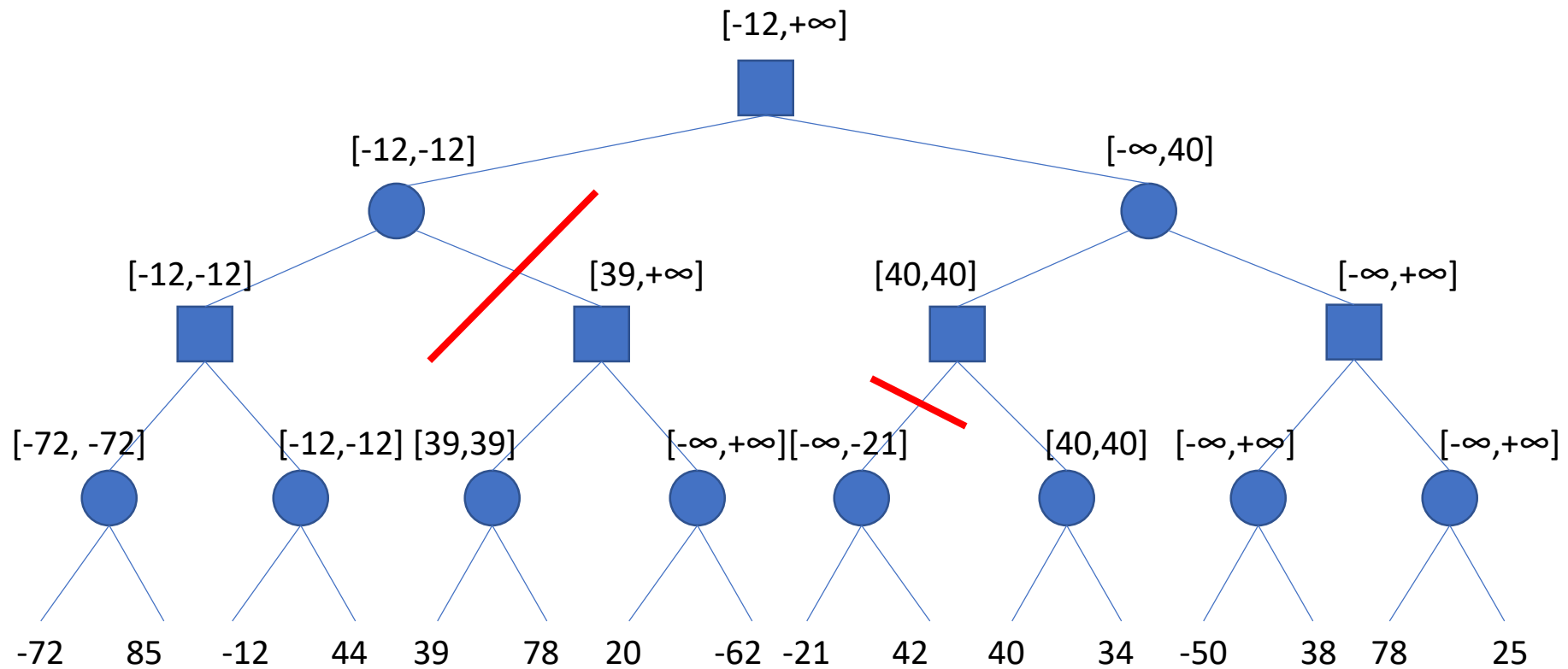
- Karena β di Min lebih kecil dari α di Max pada saat posisi di Min maka yang dipotong tertera seperti di gambar



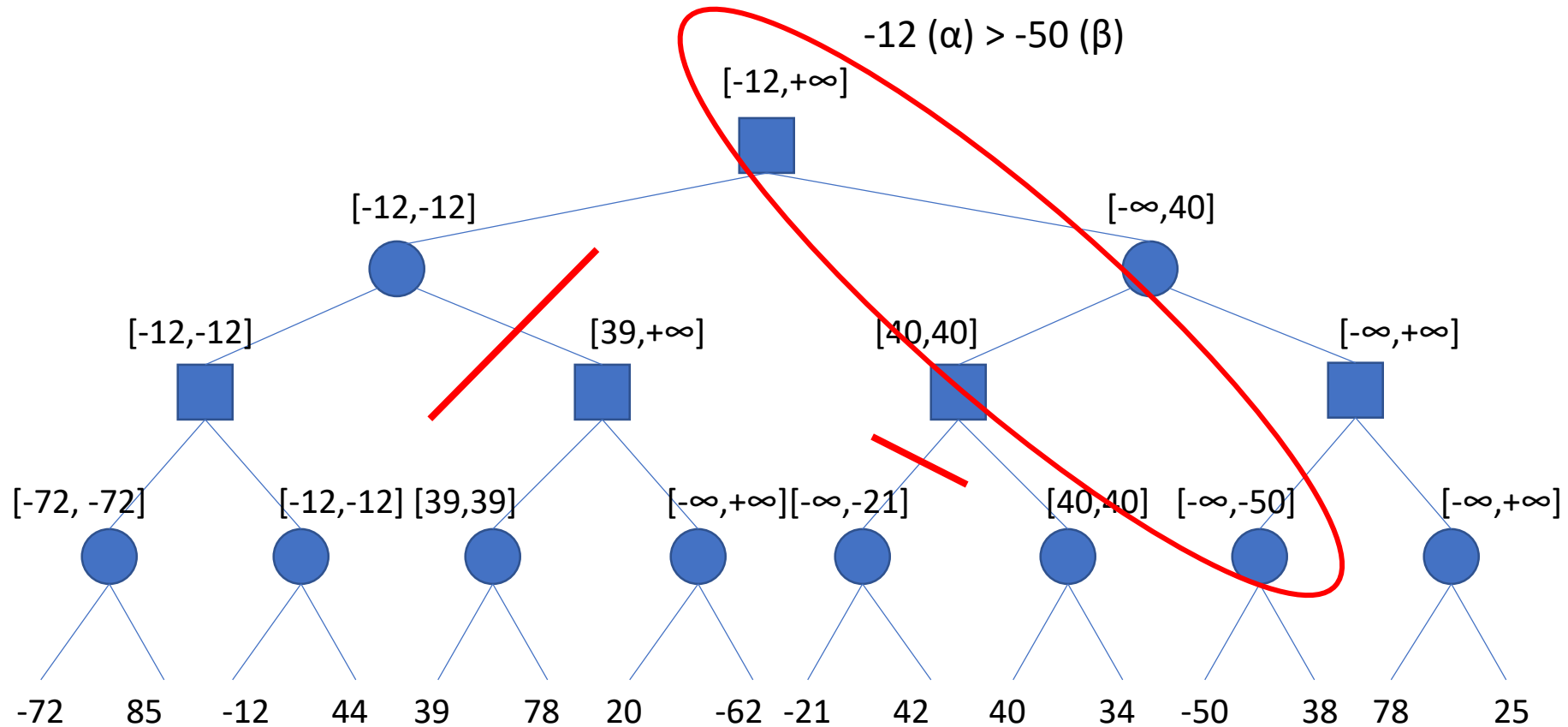
Lakukan hal yang sama seperti sebelumnya



Lakukan hal yang sama seperti sebelumnya

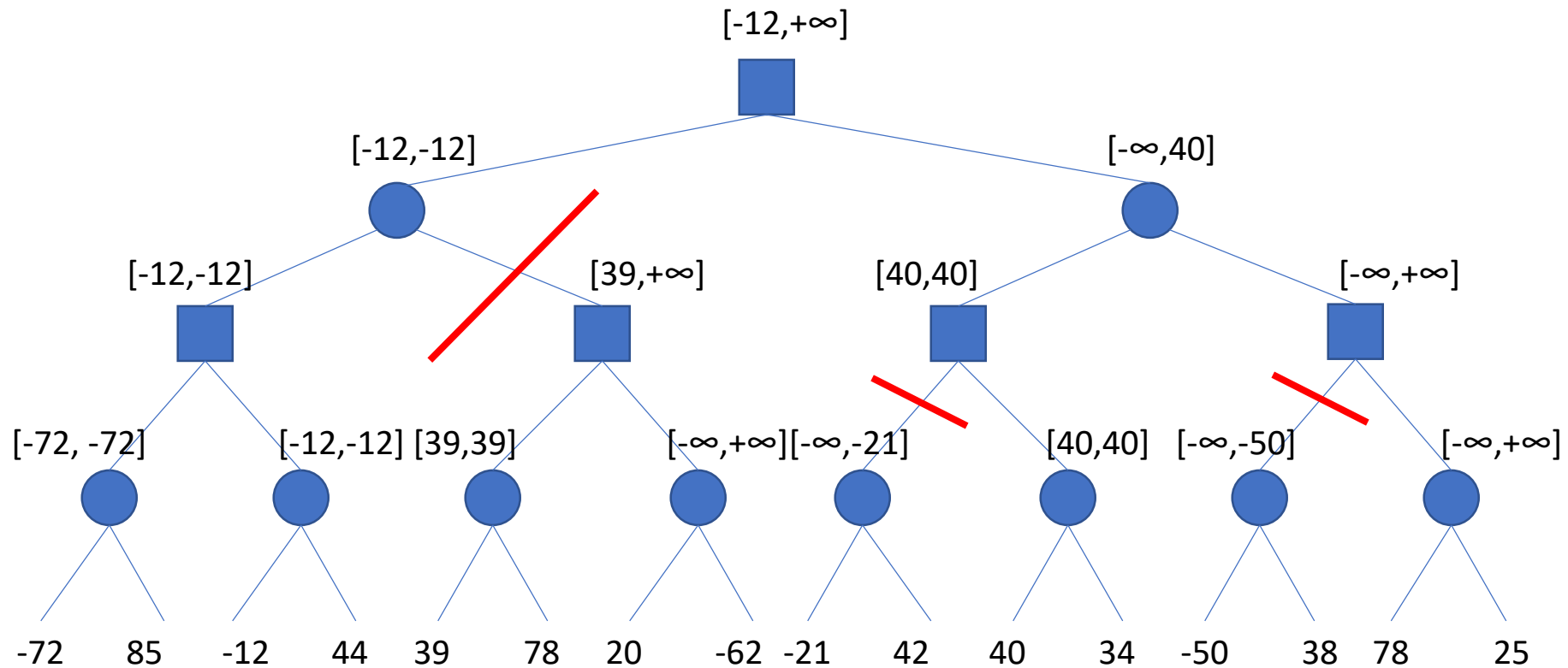


Lakukan hal yang sama seperti sebelumnya



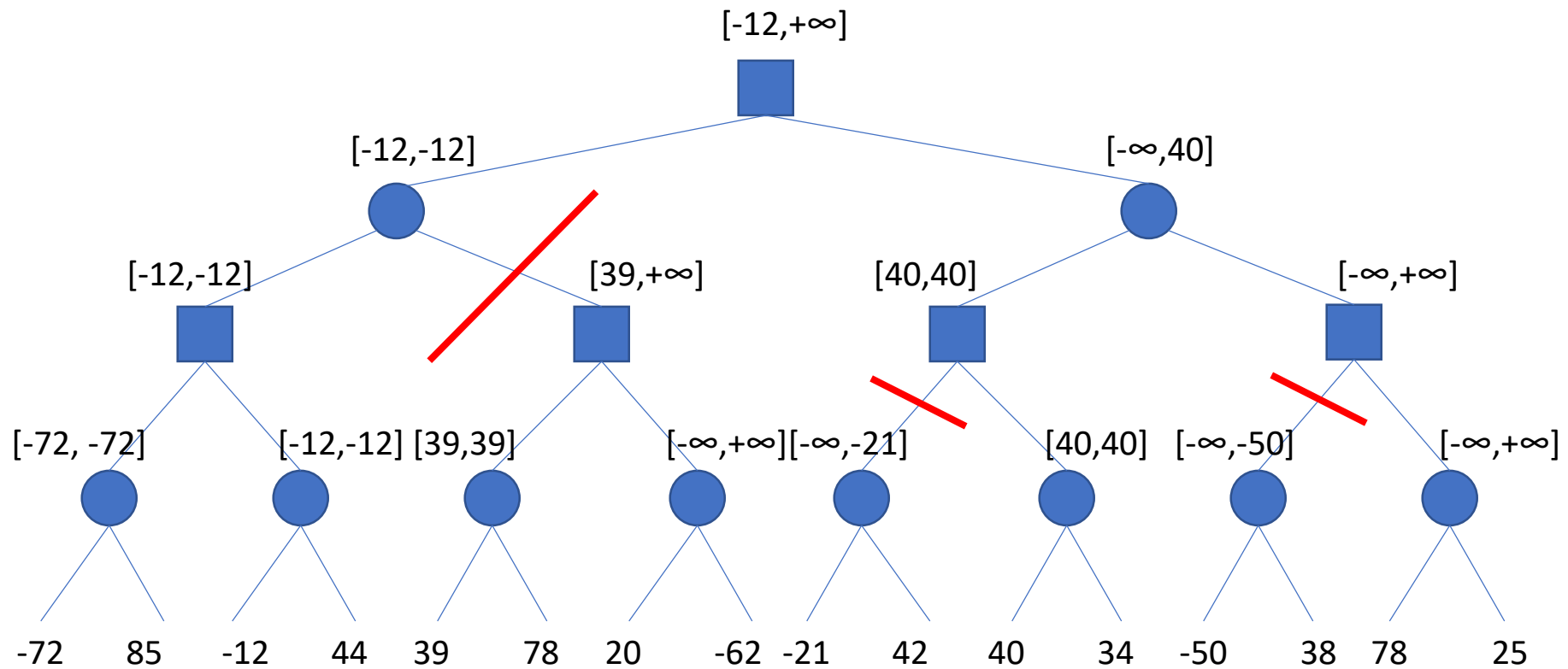
Lakukan hal yang sama seperti sebelumnya sampai kita menemukan:

- Posisi di max $\rightarrow \alpha$ di Max lebih besar dari β di Min di atasnya atau
- Posisi di min $\rightarrow \beta$ di Min lebih kecil dari α di Max yang di atasnya

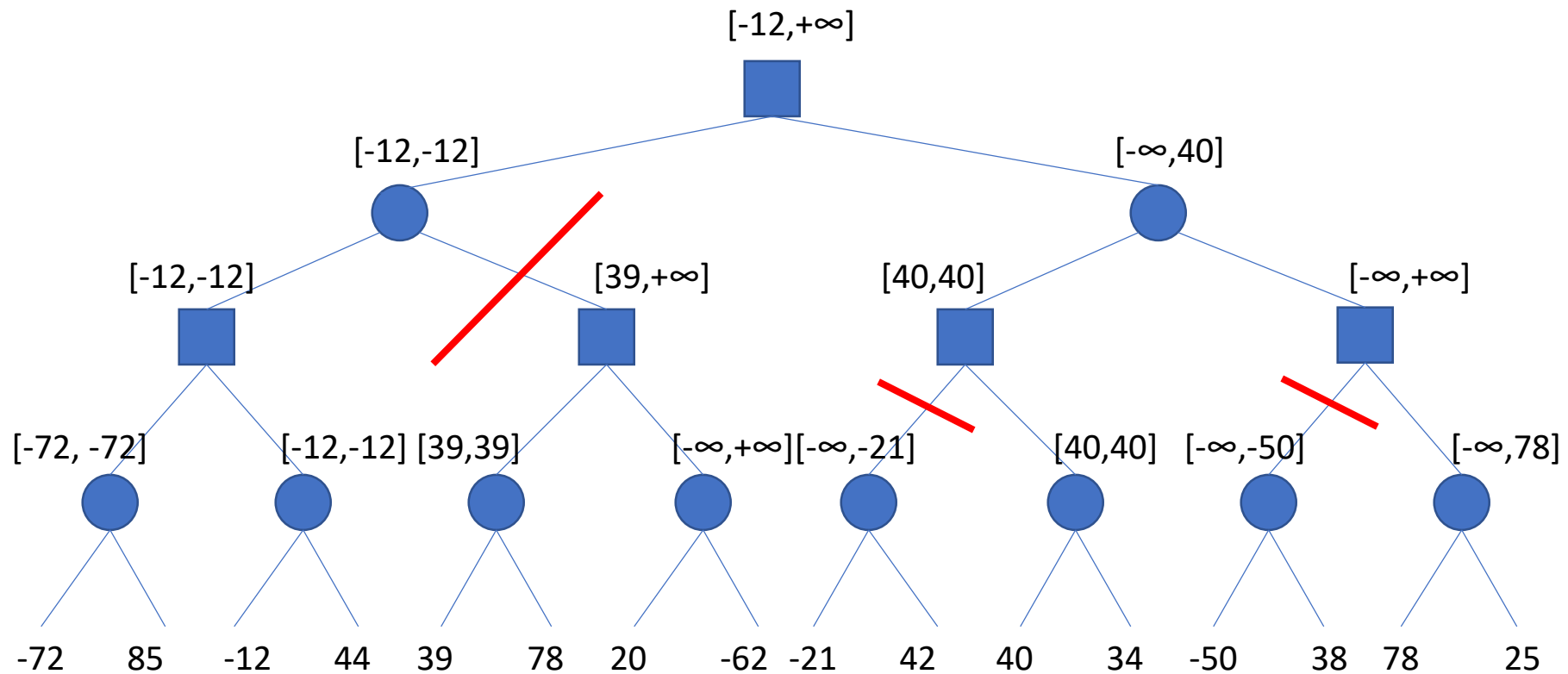


Apabila ditemukan maka potong cabang pohon yang menyambungkan Max dan Min tersebut

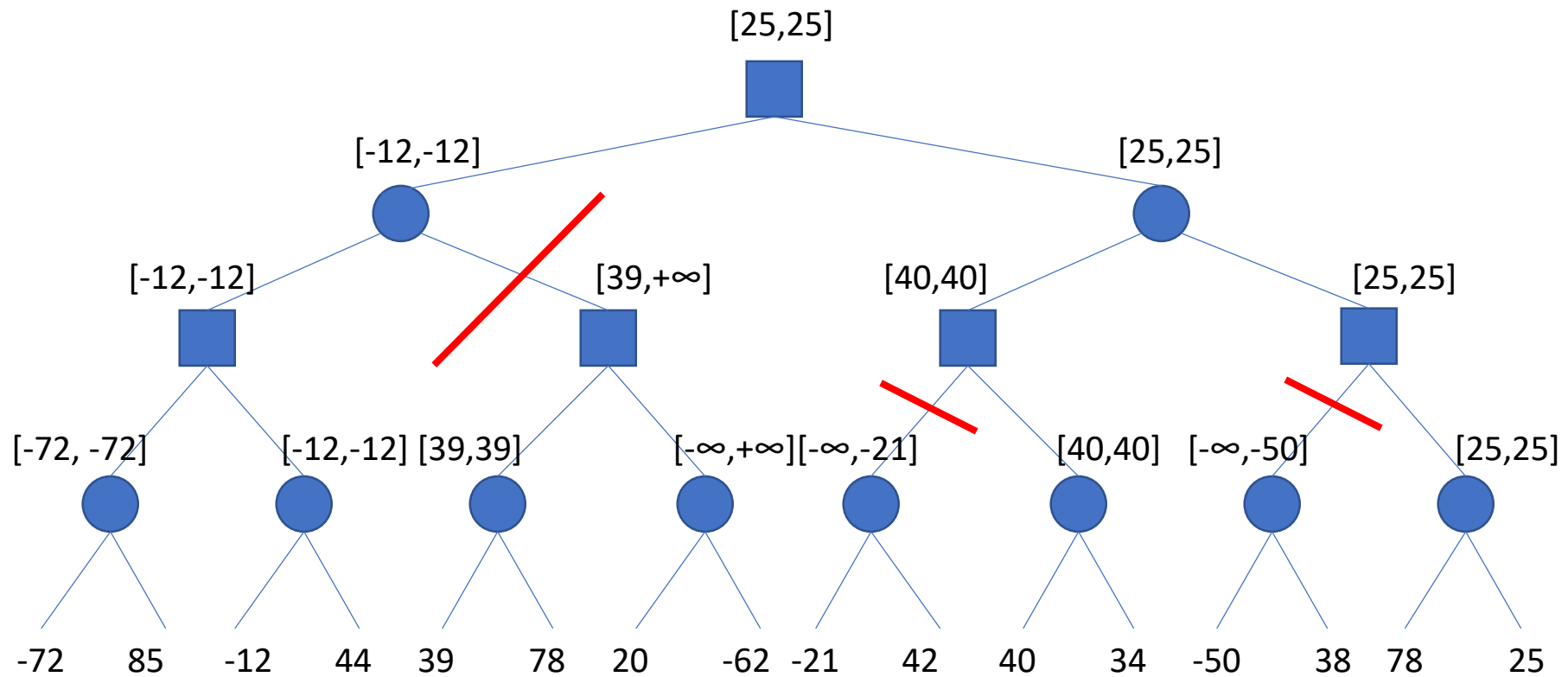
- Karena β di Min lebih kecil dari α di Max pada saat posisi di Min maka yang dipotong tertera seperti di gambar



Lakukan hal yang sama seperti sebelumnya



Lakukan hal yang sama seperti sebelumnya



Lakukan hal yang sama seperti sebelumnya