

KELOMPOK X

lelele

ALPHABETA PRUNING

KECERDASAN BUATAN F



KELOMPOK X

Line 1
Line 2
Line 3

NADIF MUSTAFA / 5025211127
FREDERICK YONATAN / 5025211121
NIZAM HAKIM / 5025211209

KECERDASAN BUATAN F



PENGERTIAN DAN TUJUAN

ALPHA-BETA PRUNING ADALAH SEBUAH ALGORITMA YANG DIGUNAKAN DALAM PERMAINAN BERBASIS POHON, SEPERTI CATUR ATAU PERMAINAN STRATEGI LAINNYA, UNTUK MENGURANGI JUMLAH SIMPUL YANG HARUS DIEVALUASI OLEH MESIN PENCARI SAAT MENCARI LANGKAH TERBAIK.

TUJUAN UTAMA DARI ALGORITMA ALPHA-BETA PRUNING ADALAH MENGURANGI WAKTU KOMPUTASI YANG DIPERLUKAN UNTUK MENGEVALUASI SIMPUL-SIMPUL DALAM POHON PERMAINAN. INI DICAPAI DENGAN MEMOTONG ATAU MENGEVALUASI SIMPUL-SIMPUL YANG TIDAK PERLU DIEVALUASI LEBIH LANJUT, KARENA SIMPUL-SIMPUL TERSEBUT TIDAK AKAN MEMPENGARUHI KEPUTUSAN AKHIR YANG DIAMBIL.



LANGKAH - LANGKAH



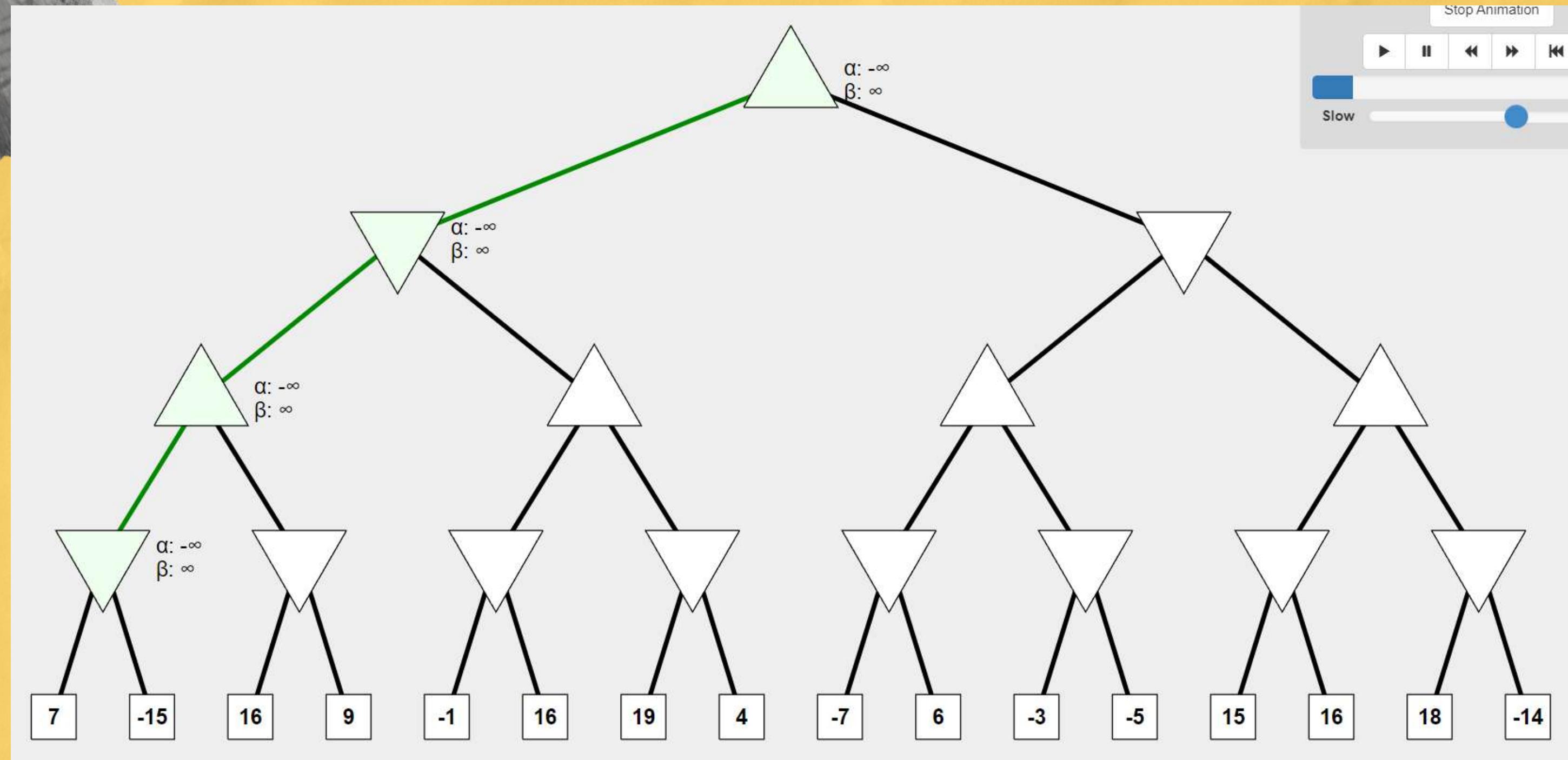
Berikut adalah langkah-langkah umum dalam algoritma alpha-beta pruning :

- Representasikan permainan sebagai pohon: Posisi awal permainan adalah akar pohon, dan setiap langkah yang mungkin dari posisi tersebut adalah anak-anak dari simpul tersebut. Pohon ini akan memiliki cabang yang mewakili setiap langkah yang mungkin dalam permainan.
- Evaluasi pohon menggunakan pencarian minimax: Setiap simpul di pohon diberi skor evaluasi berdasarkan pemain yang sedang bergerak. Pada setiap tingkat kedalaman, pemain maksimum mencoba memaksimalkan skor, sedangkan pemain minimum meminimalkan skor. Nilai-nilai ini dipropagaskan ke atas dan ke bawah pohon dengan menggunakan algoritma minimax.
- Gunakan alpha dan beta untuk memotong simpul yang tidak relevan: Alpha dan beta adalah batasan atas dan batasan bawah pada nilai yang diharapkan dari simpul. Pada awalnya, alpha diatur sebagai negatif tak terbatas dan beta diatur sebagai positif tak terbatas. Ketika algoritma bergerak ke bawah pohon, nilai alpha dan beta diperbarui berdasarkan langkah-langkah yang dievaluasi. Jika pada suatu titik alpha lebih besar atau sama dengan beta, pencarian pada cabang tersebut dihentikan, karena tidak ada gunanya mengevaluasi simpul-simpul anak yang lain.

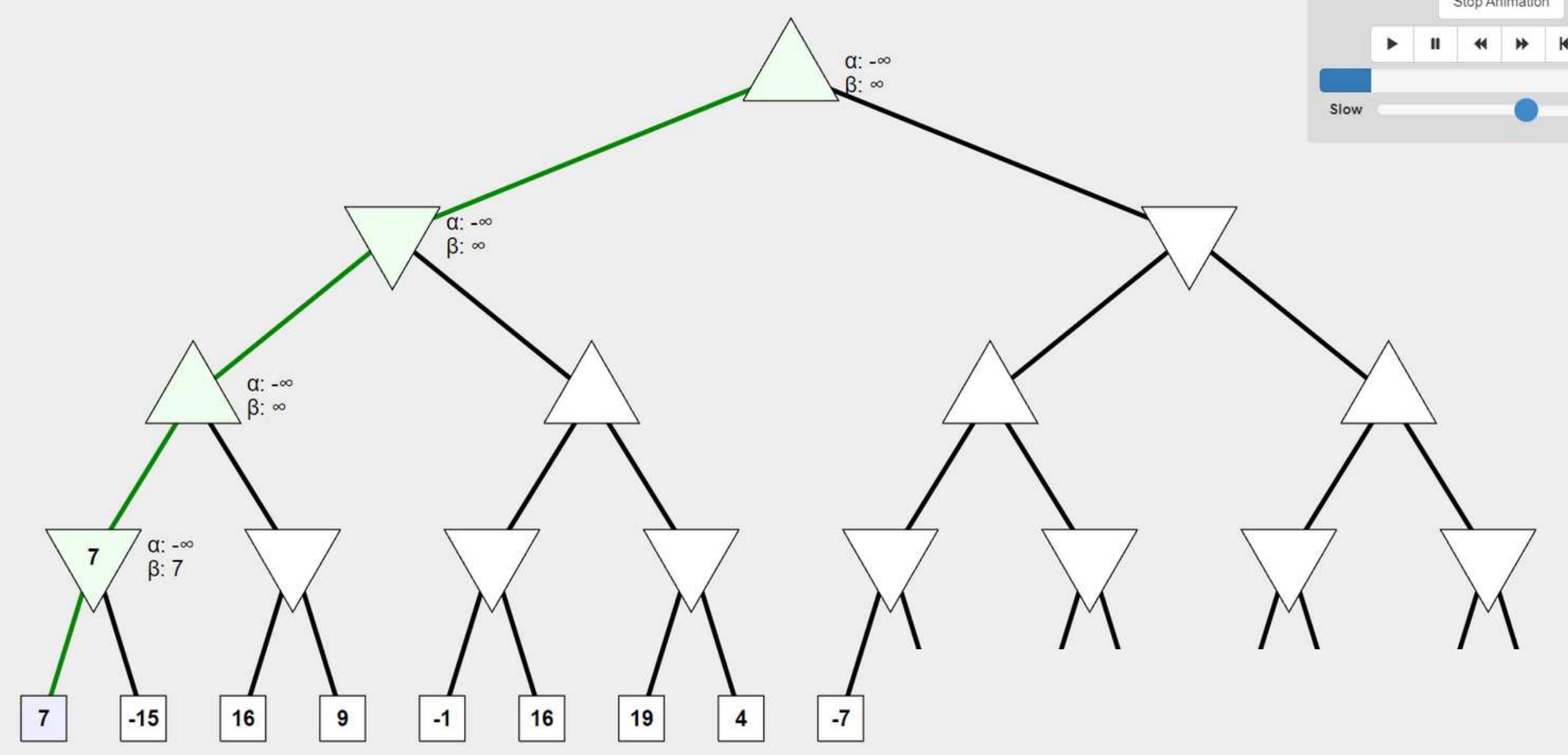
- Pemangkasan alpha-beta: Pemangkasan dilakukan saat pencarian bergerak ke atas pohon. Ketika simpul selesai dievaluasi, nilai alpha atau beta diwariskan ke simpul induk sesuai dengan peran pemain. Pemain maksimum memperbarui alpha jika nilai simpul saat ini lebih besar dari alpha, sementara pemain minimum memperbarui beta jika nilai simpul saat ini lebih kecil dari beta. Dengan pemangkasan ini, simpul-simpul yang berada di atasnya tidak perlu dievaluasi lebih lanjut.
- Teruskan pencarian hingga ke tingkat yang diinginkan: Pencarian dilanjutkan hingga tingkat yang diinginkan dalam pohon, atau hingga batasan waktu atau sumber daya lainnya tercapai.

ILUSRASI ALPHA-BETA PRUNING



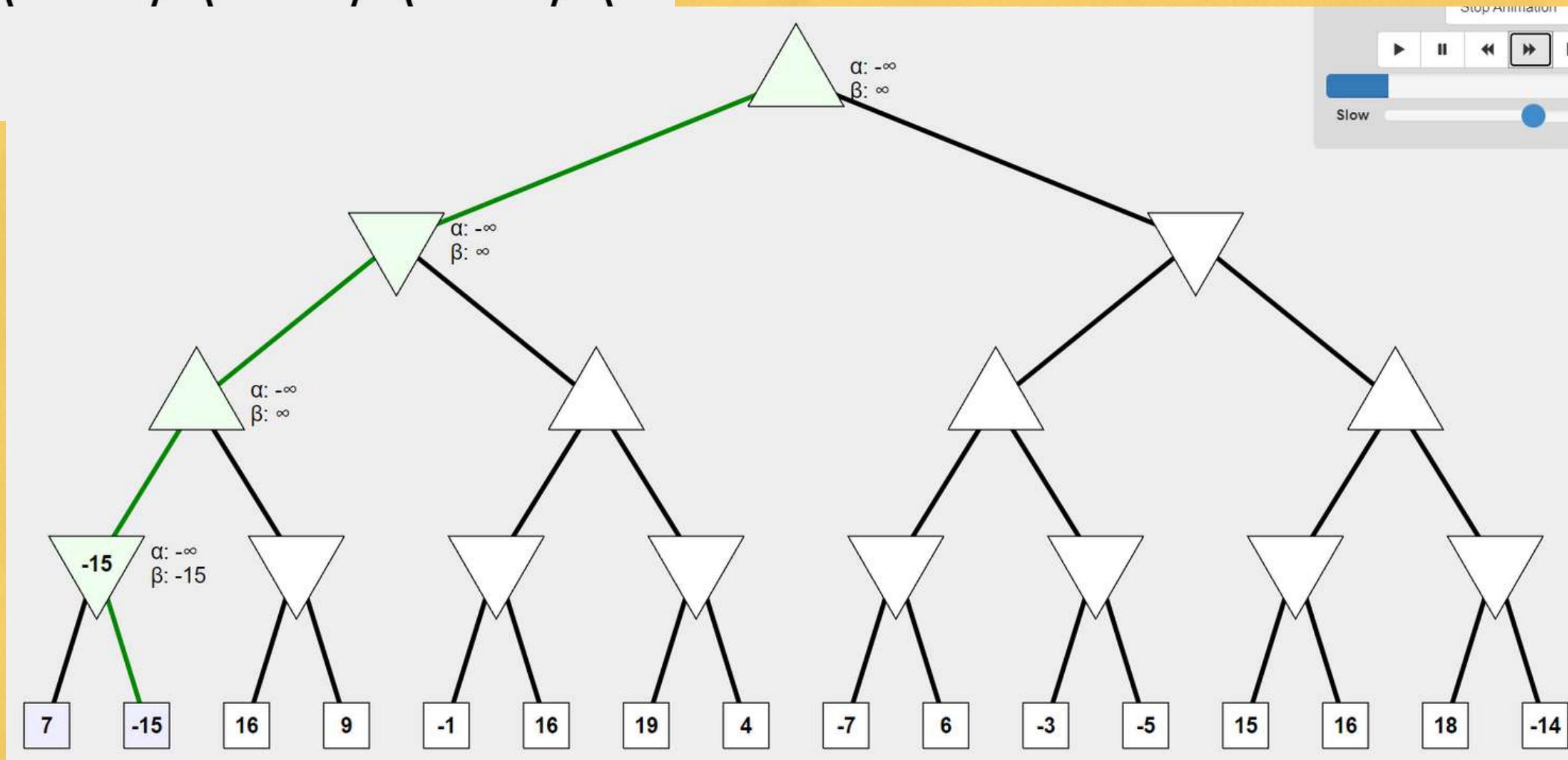


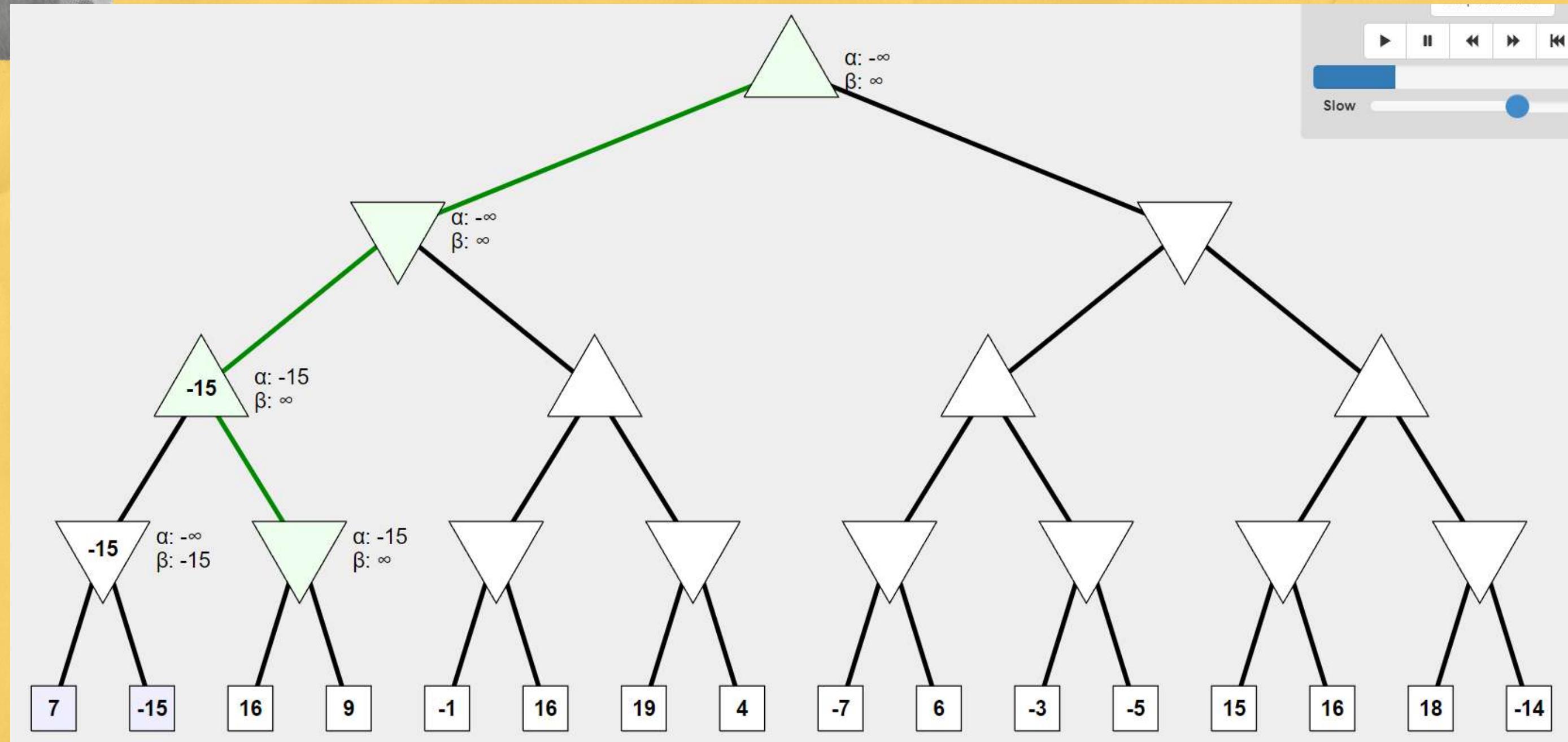
- Misalkan segitiga atas = putih, segitiga bawah = hitam, pada permainan catur
- Inisiasi nilai $\alpha = -\infty$ dan $\beta = \infty$
- α untuk meyimpan nilai maksimum yang bisa didapat putih
- β untuk menyimpan nilai minimum yang bisa didapat hitam



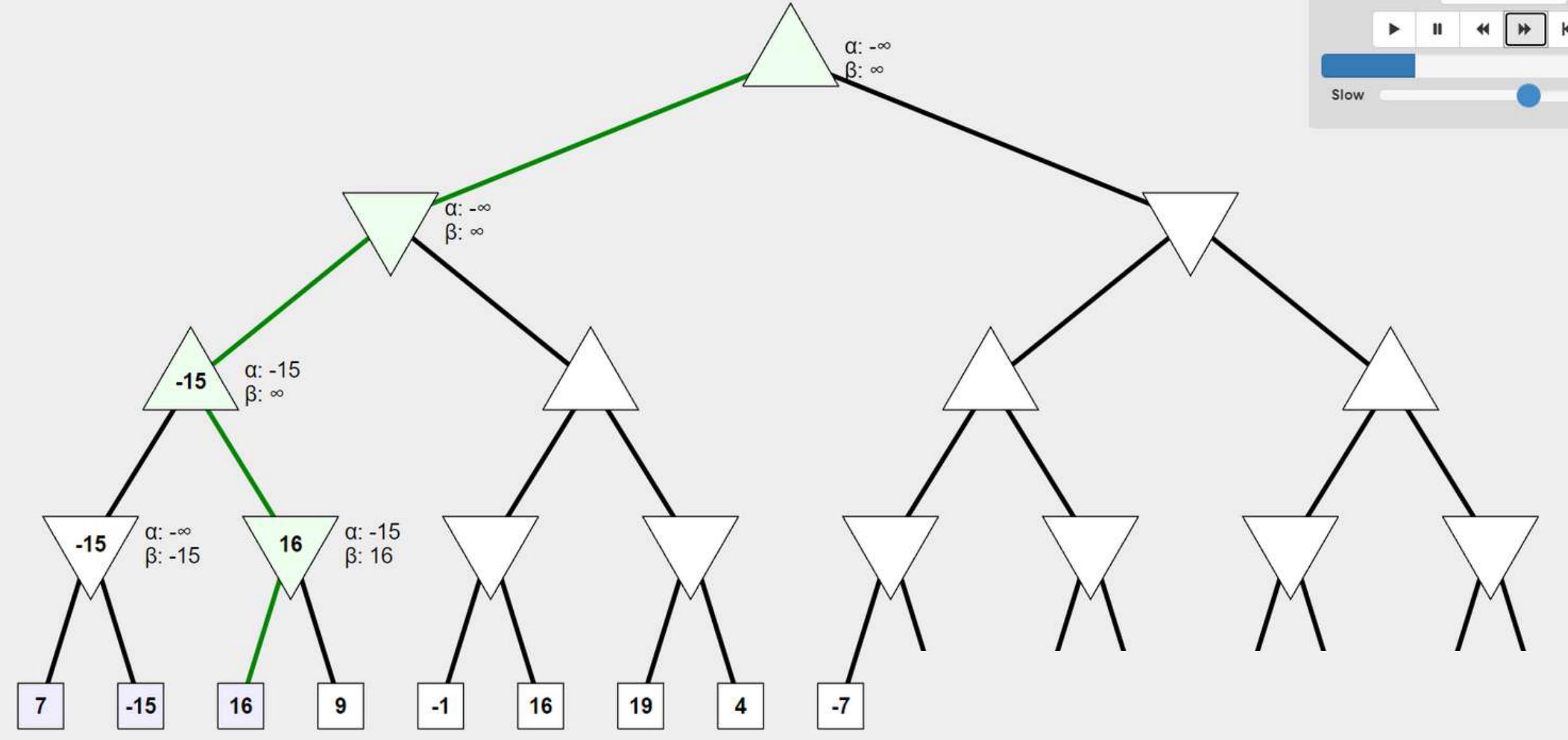
- Dimulai dari leaf paling kiri.
- Karena nilai dalam hitam masih kosong, maka 7 dipilih secara langsung.
- Lalu 7 dibandingkan dengan nilai β sekarang yaitu ∞ , lalu β diubah menjadi 7 karena $7 < \infty$.

- Kemudian melanjutkan ke leaf kanan
- Karena saat ini adalah move hitam, maka memilih nilai minimum antara 7 dengan -15
- Sehingga dipilih -15 dan β diganti menjadi -15



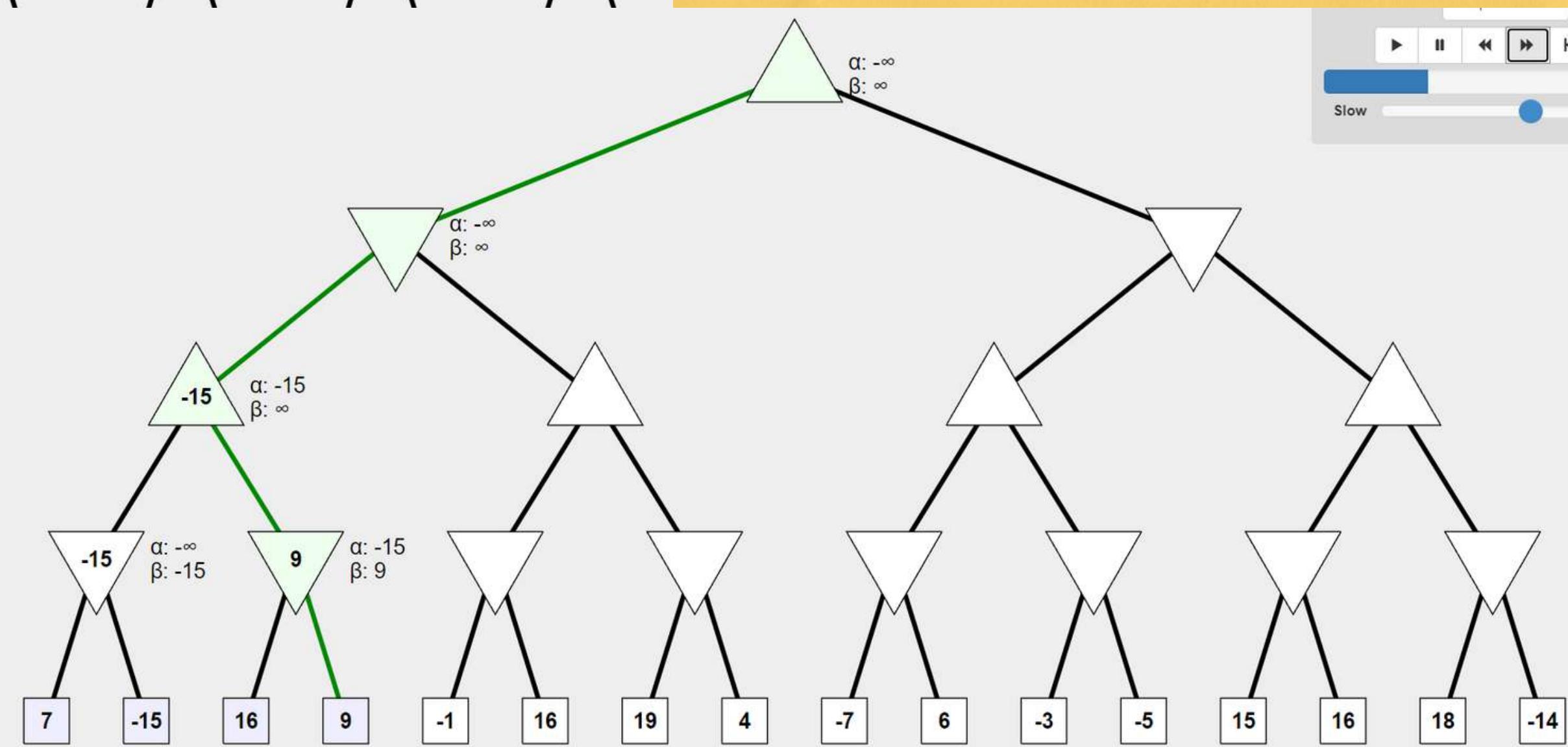


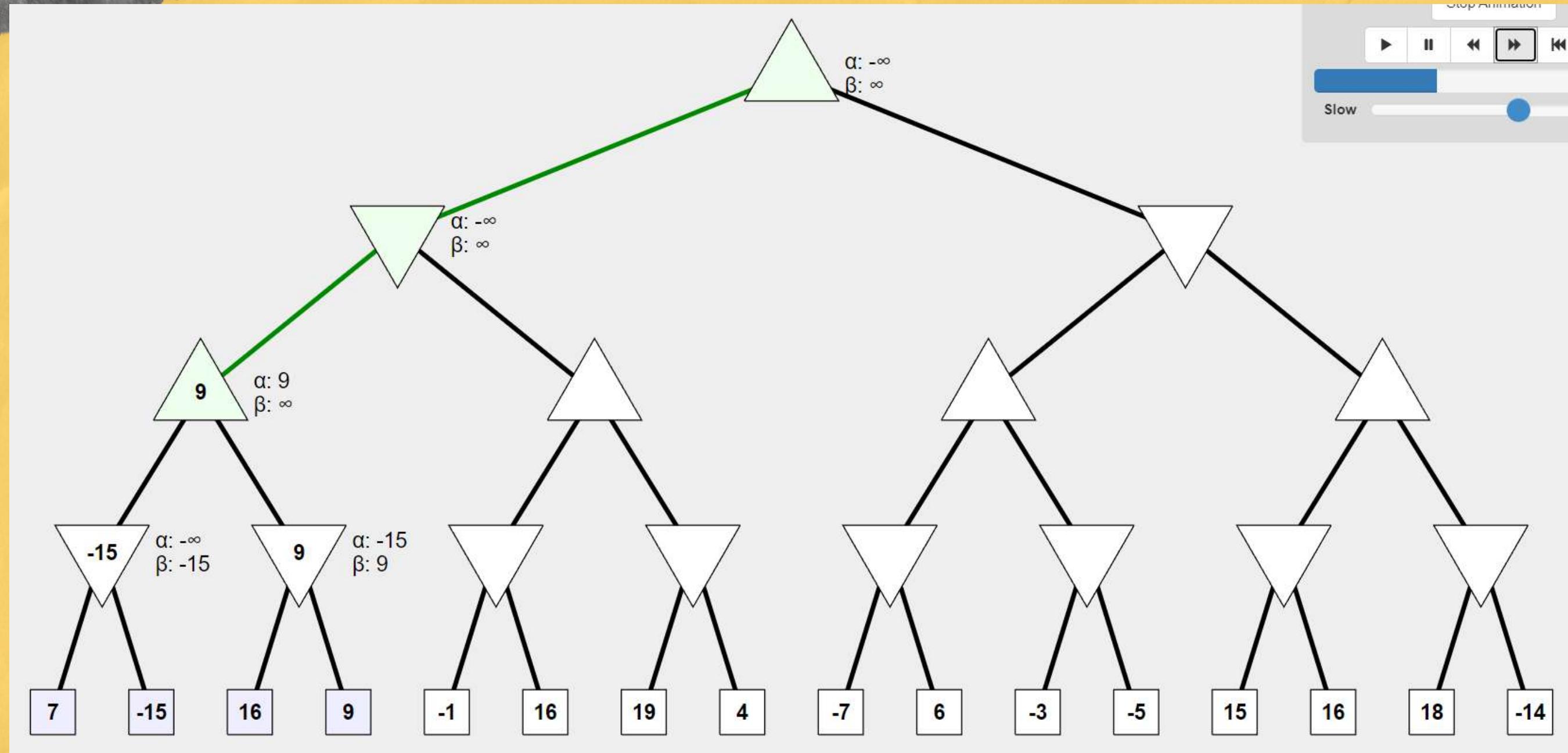
- Dilanjutkan ke tingkat di atasnya yaitu move putih
- Karena nilai yang sudah ditemukan hanya nilai pada child kiri = -15 maka -15 dipilih menjadi nilai maksimum
- α diubah menjadi -15 karena $-15 > -\infty$
- Dilanjutkan mencari nilai untuk child kanan



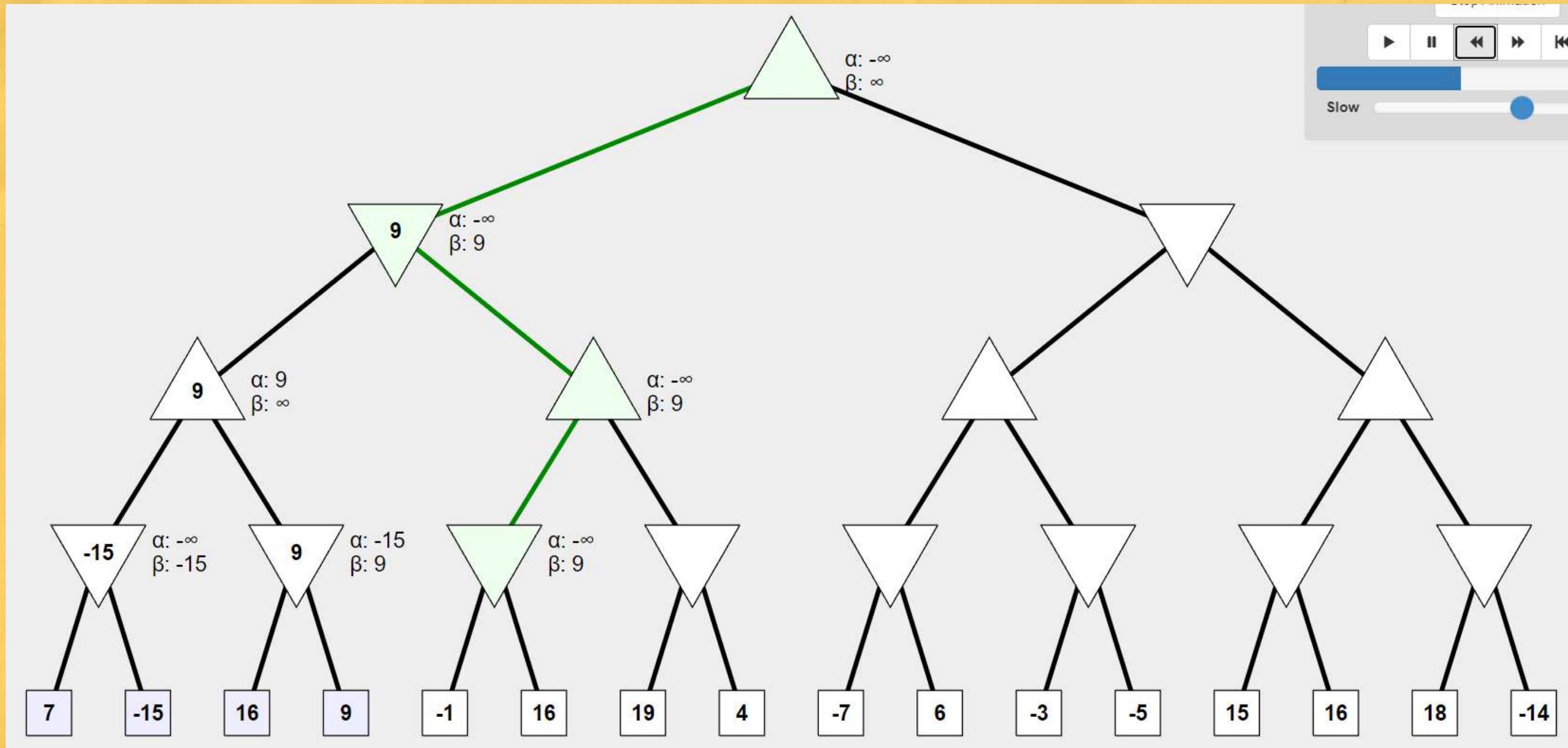
- Dimulai dari leaf kiri.
- Karena nilai dalam hitam masih kosong, maka 16 dipilih secara langsung.
- Lalu 16 dibandingkan dengan nilai β sekarang yaitu ∞ , lalu β diubah menjadi 16 karena $16 < \infty$.

- Kemudian melanjutkan ke leaf kanan
- Karena saat ini adalah move hitam, maka memilih nilai minimum antara 16 dengan 9.
- Sehingga dipilih 9 dan β diganti menjadi 9.



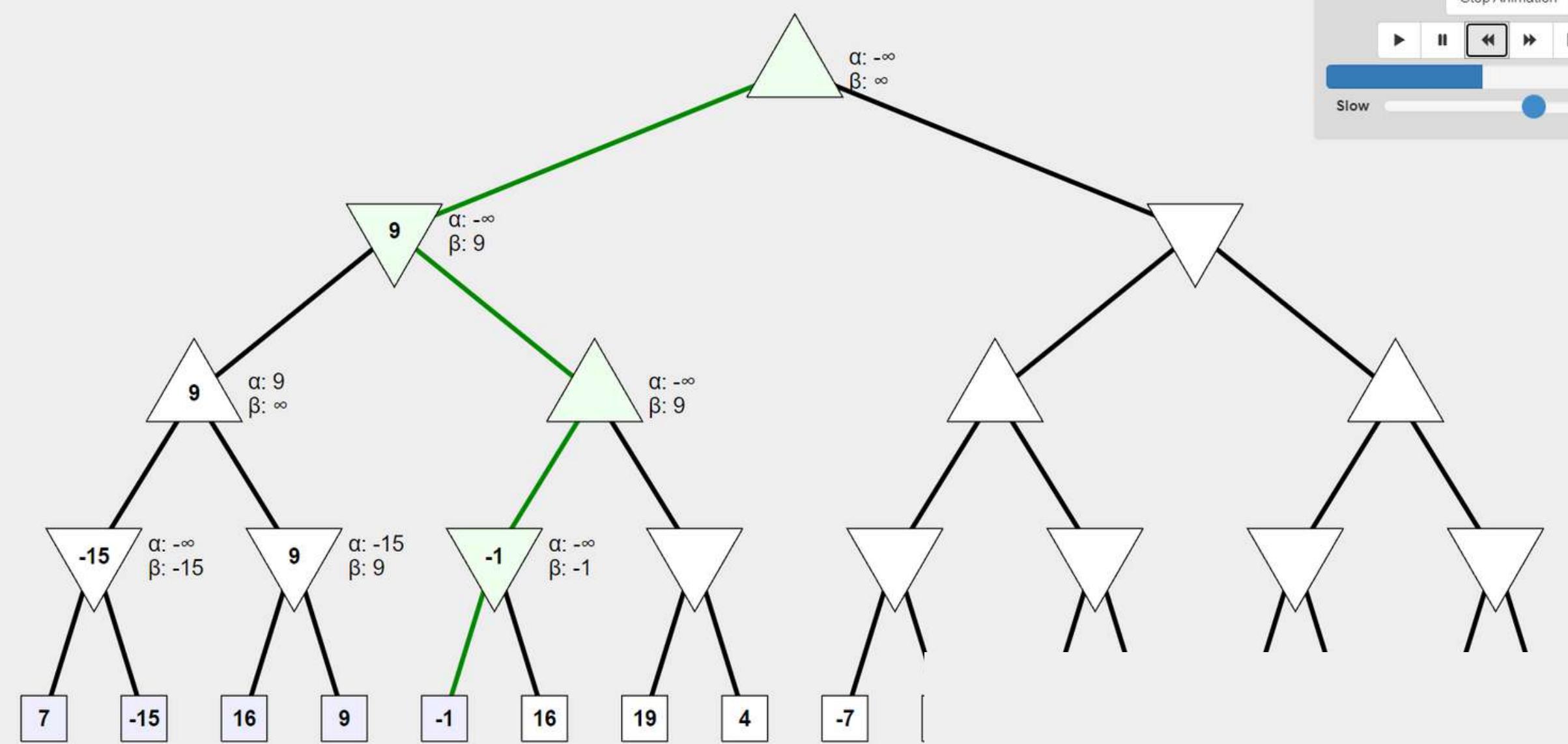


- Kembali ke tingkat diatasnya yaitu move putih
- Sekarang nilai dari child kanan sudah ditemukan yaitu 9 sehingga memilih nilai maksimum antara -15 dan 9 yaitu 9
- α diubah menjadi 9 karena $9 > -15$
- Dilanjutkan ke tingkat diatasnya

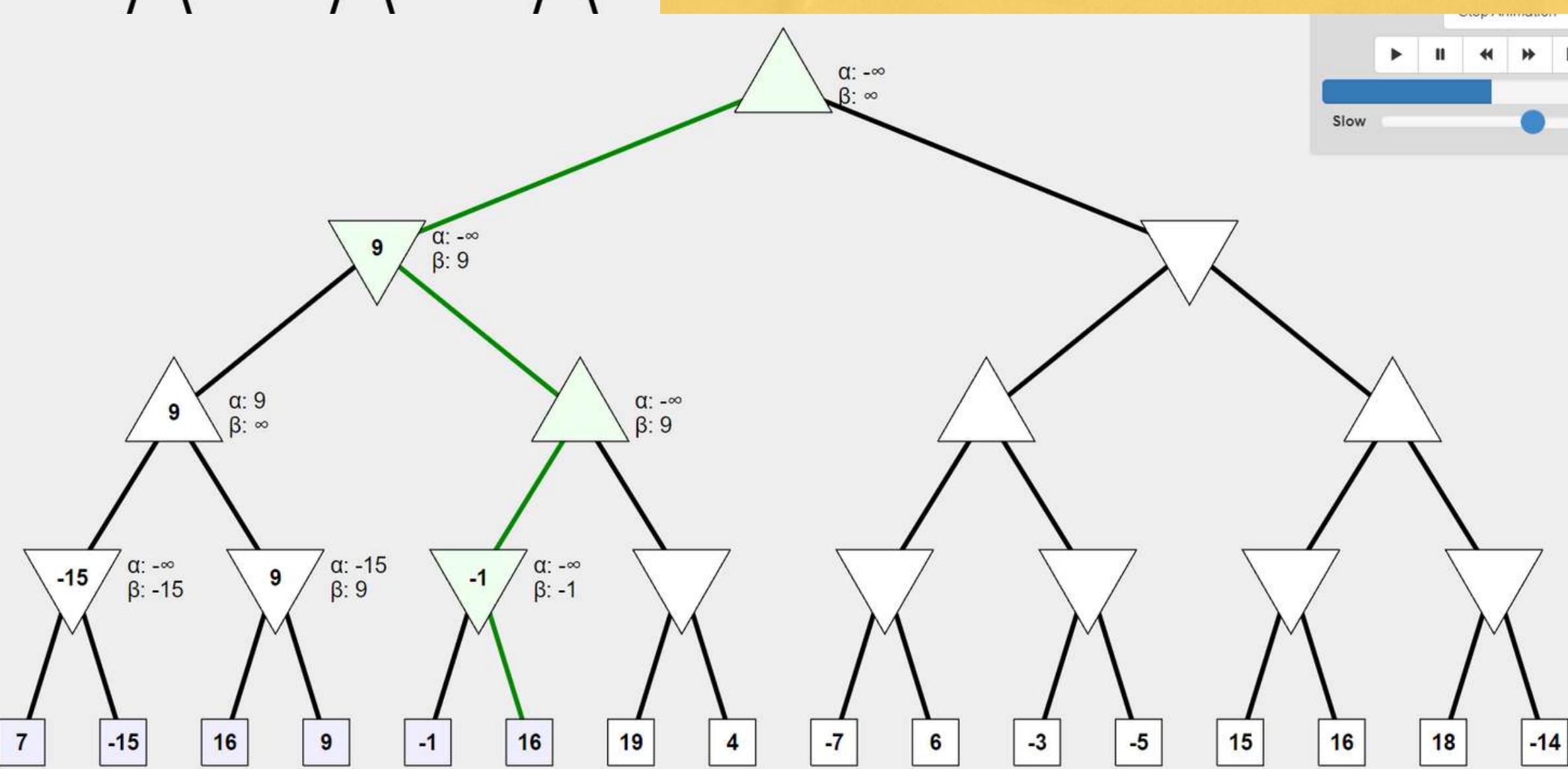


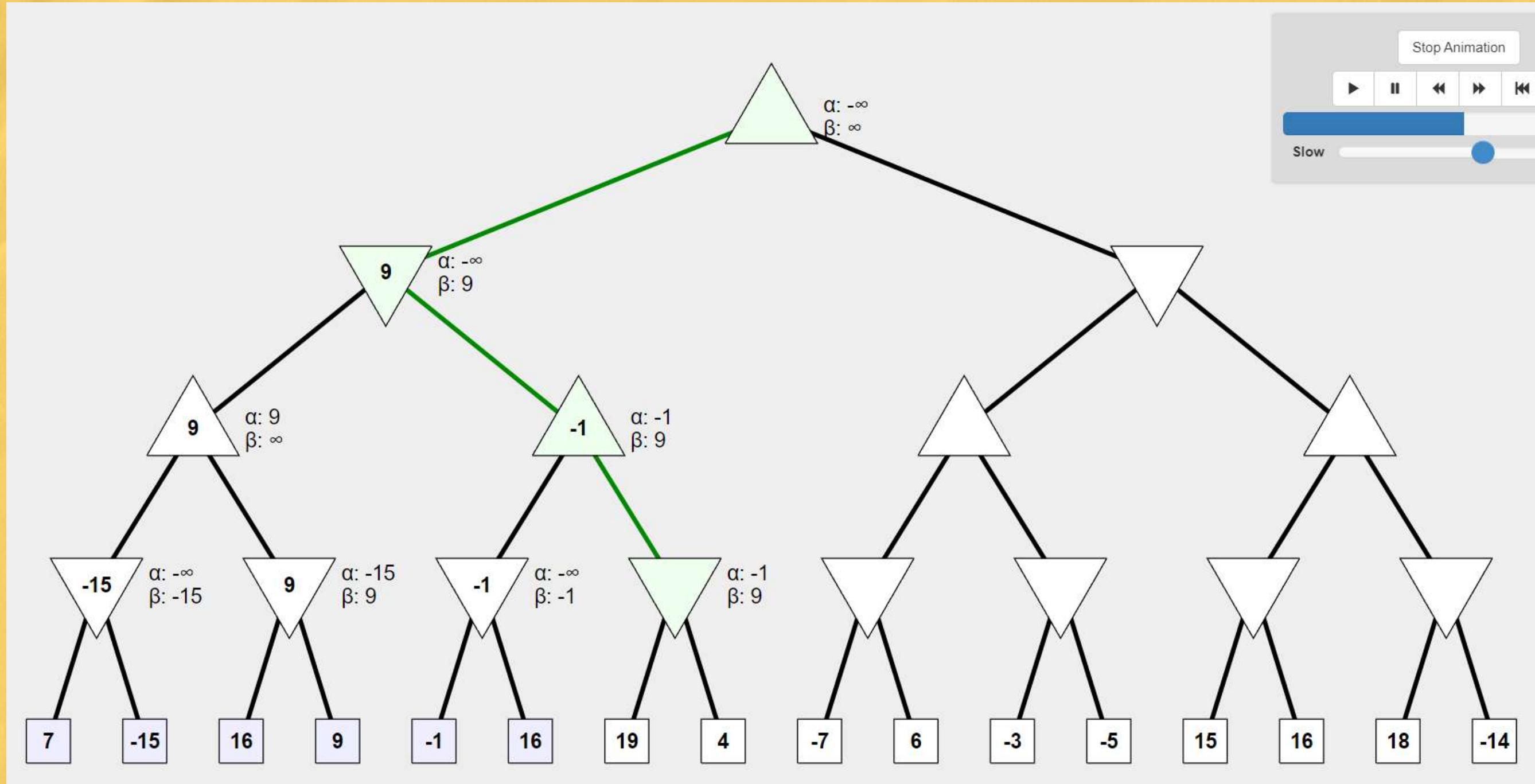
- Dilanjutkan ke tingkat di atasnya yaitu move hitam.
- Karena nilai yang sudah ditemukan hanya nilai pada child kiri = 9 maka 9 dipilih menjadi nilai minimum.
- β diubah menjadi 9 karena $9 < \infty$
- Dilanjutkan mencari nilai untuk child kanan.

- Dimulai dari leaf kiri.
- Karena nilai dalam hitam masih kosong, maka -1 dipilih secara langsung.
- Lalu -1 dibandingkan dengan nilai β sekarang yaitu 9, lalu β diubah menjadi -1 karena $-1 < 9$.



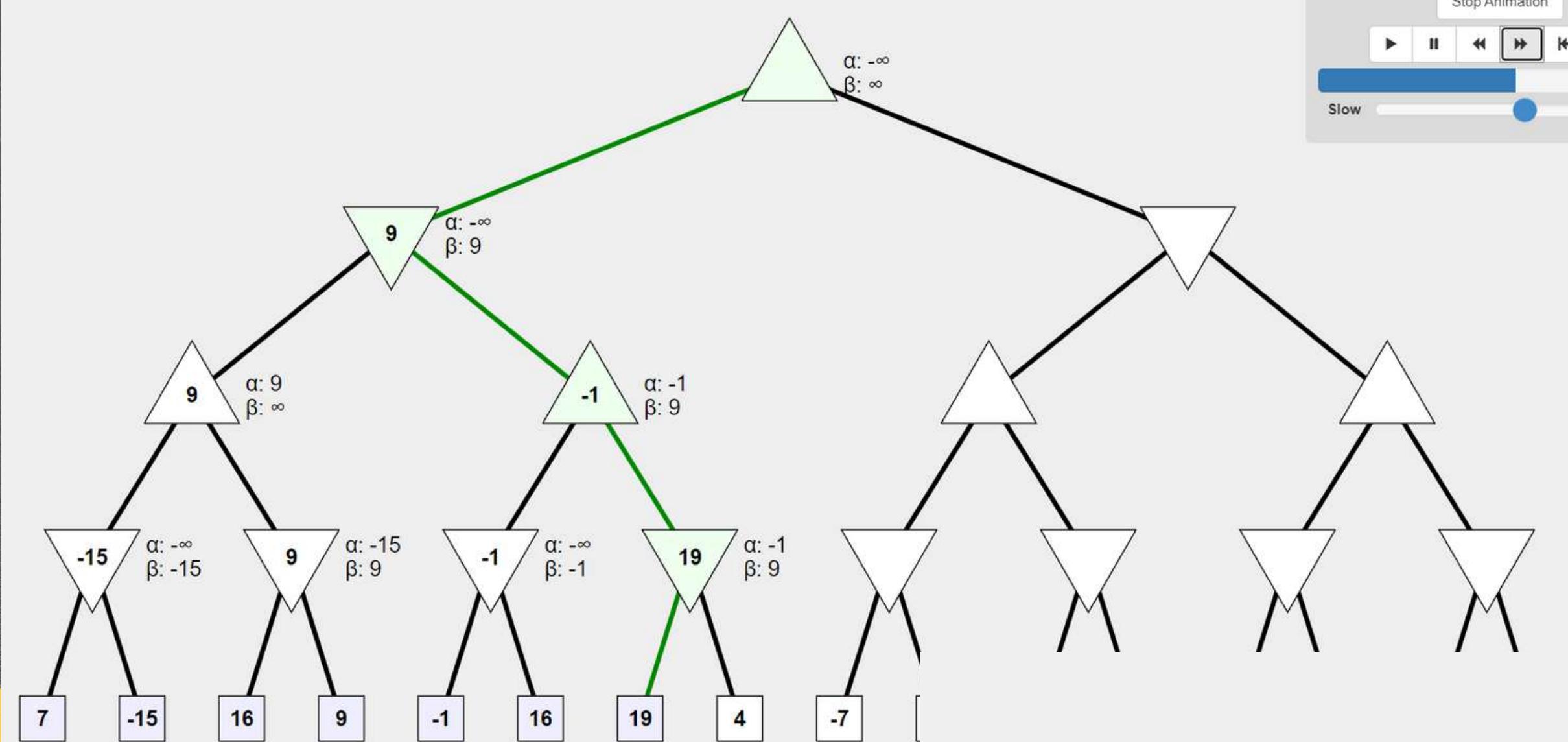
- Kemudian melanjutkan ke leaf kanan
- Karena saat ini adalah move hitam, maka memilih nilai minimum antara -1 dengan 16.
- Sehingga dipilih -1 dan β tetap -1.



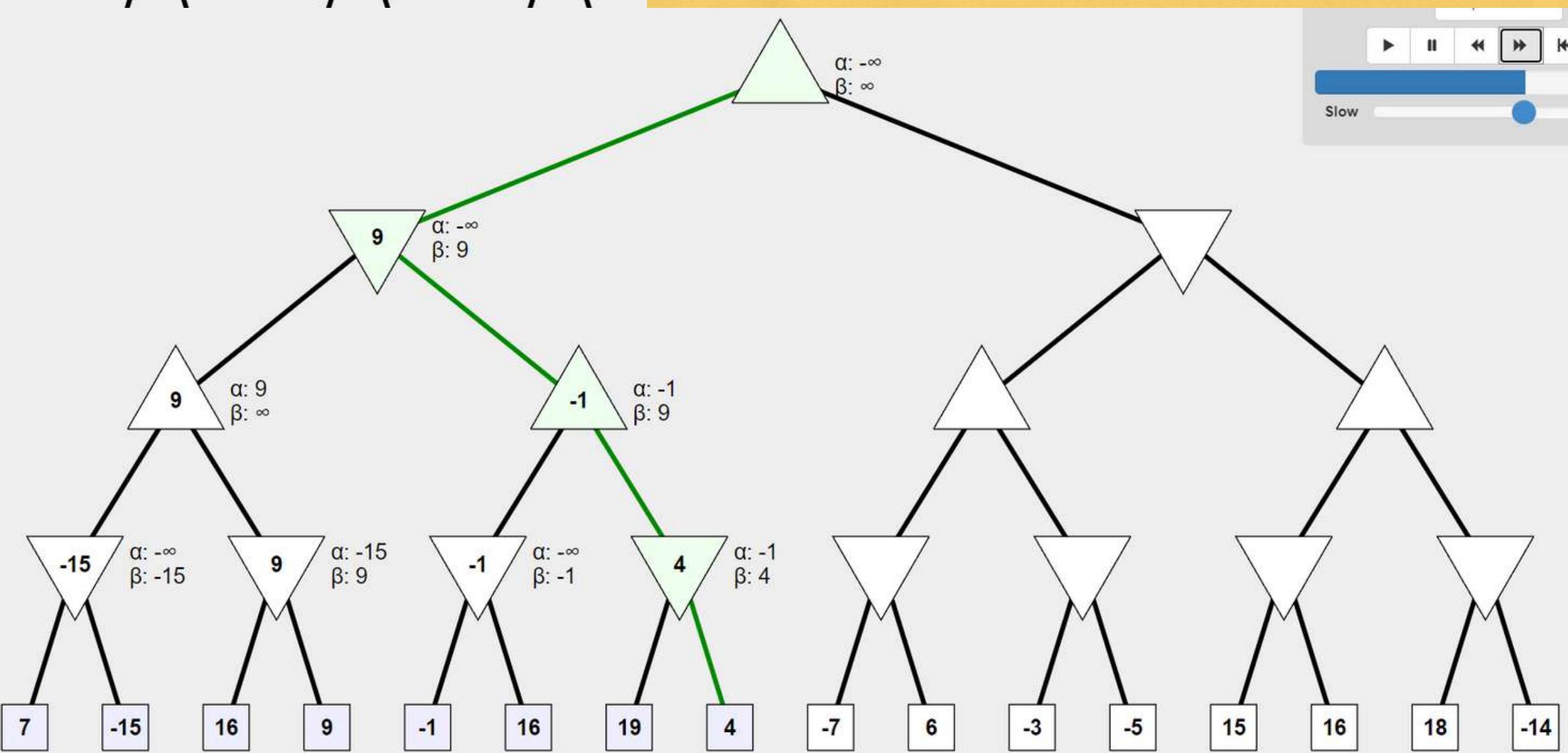


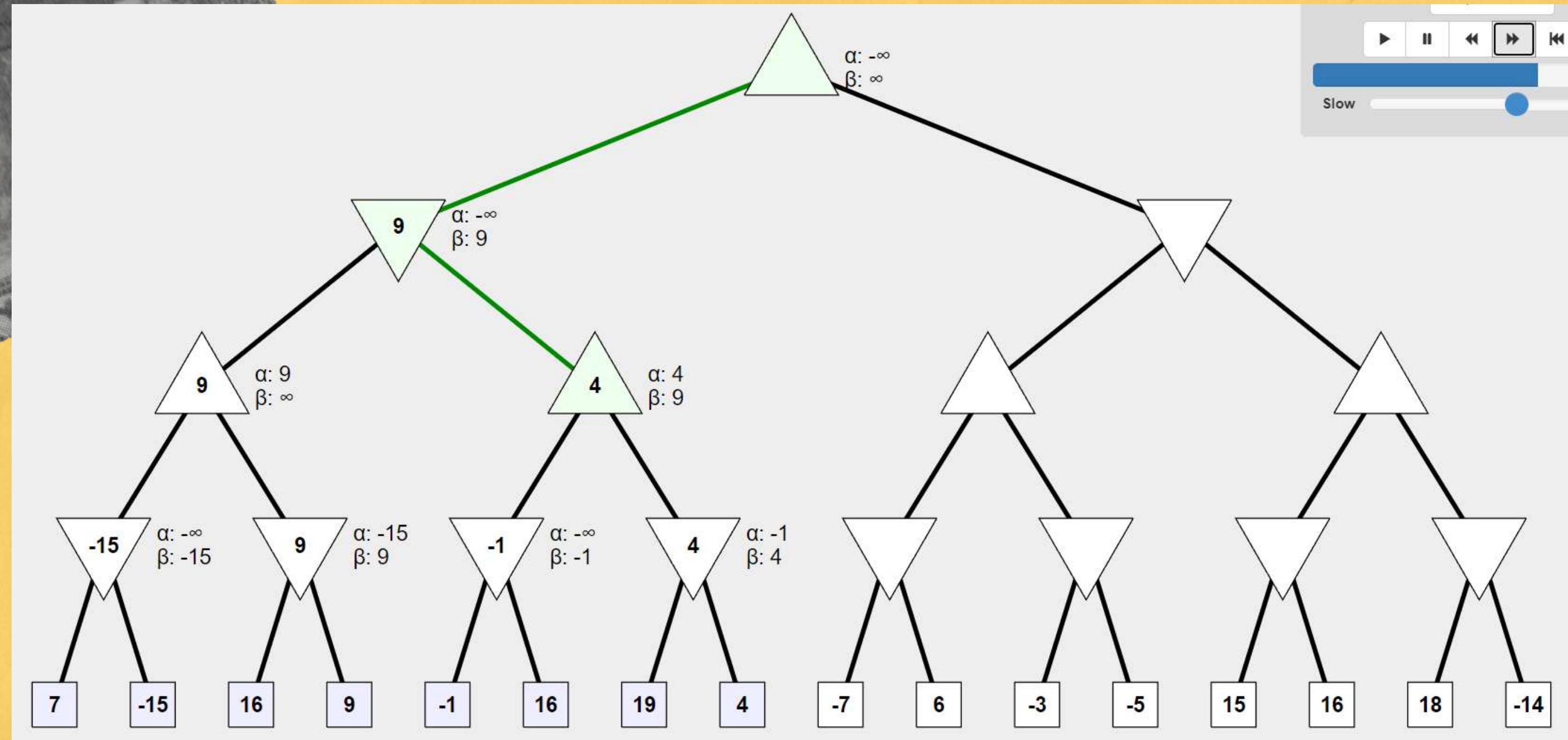
- Dilanjutkan ke tingkat di atasnya yaitu move putih
- Karena nilai yang sudah ditemukan hanya nilai pada child kiri = -1 maka -1 dipilih menjadi nilai maksimum
- α diubah menjadi -1 karena $-1 > -\infty$
- Dilanjutkan mencari nilai untuk child kanan

- Dimulai dari leaf kiri.
- Karena nilai dalam hitam masih kosong, maka 19 dipilih secara langsung.
- Lalu 19 dibandingkan dengan nilai β sekarang yaitu 9, lalu β tetap bernilai 9 karena $9 < 19$.

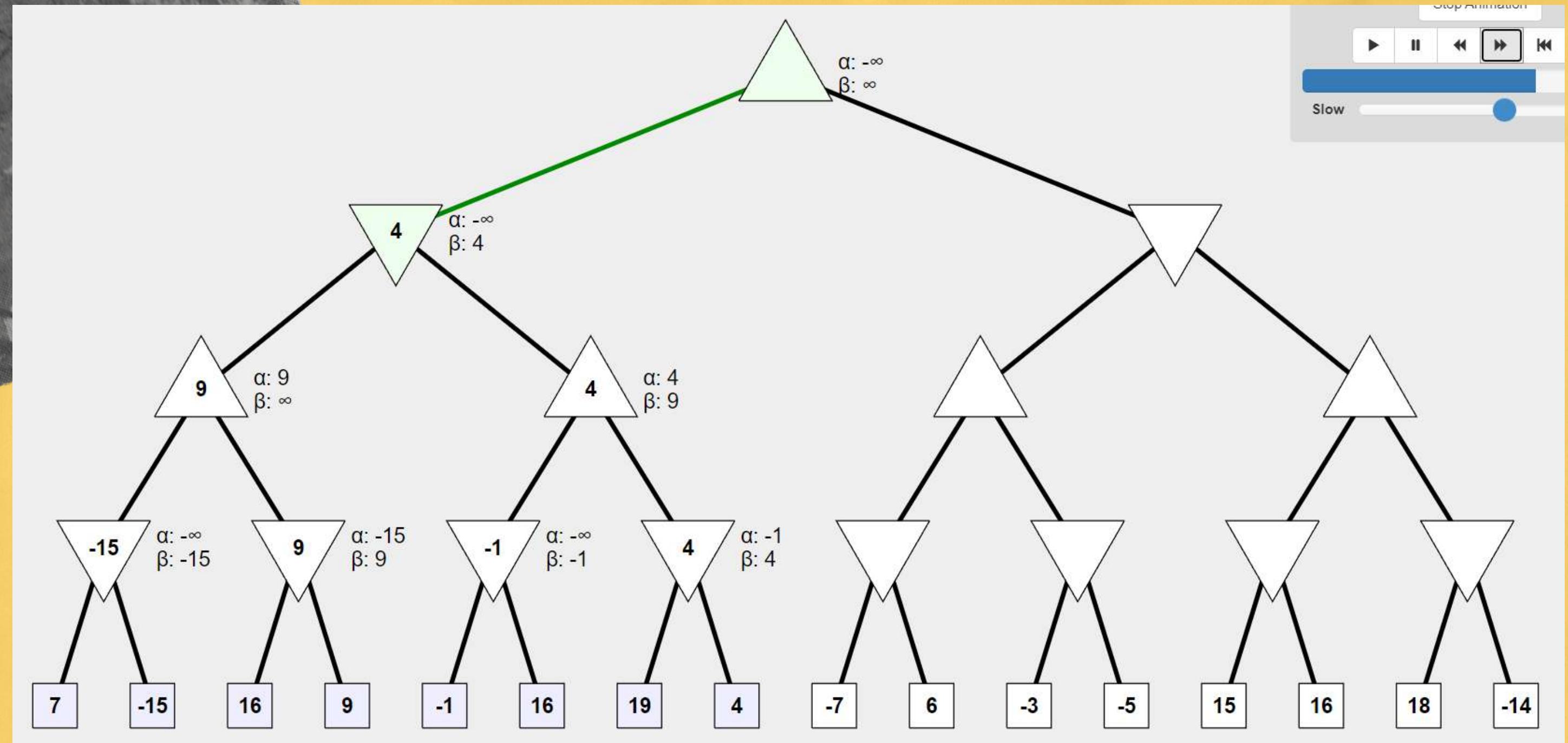


- Kemudian melanjutkan ke leaf kanan
- Karena saat ini adalah move hitam, maka memilih nilai minimum antara 4 dengan 19.
- Sehingga dipilih 4 dan β diubah menjadi 4.

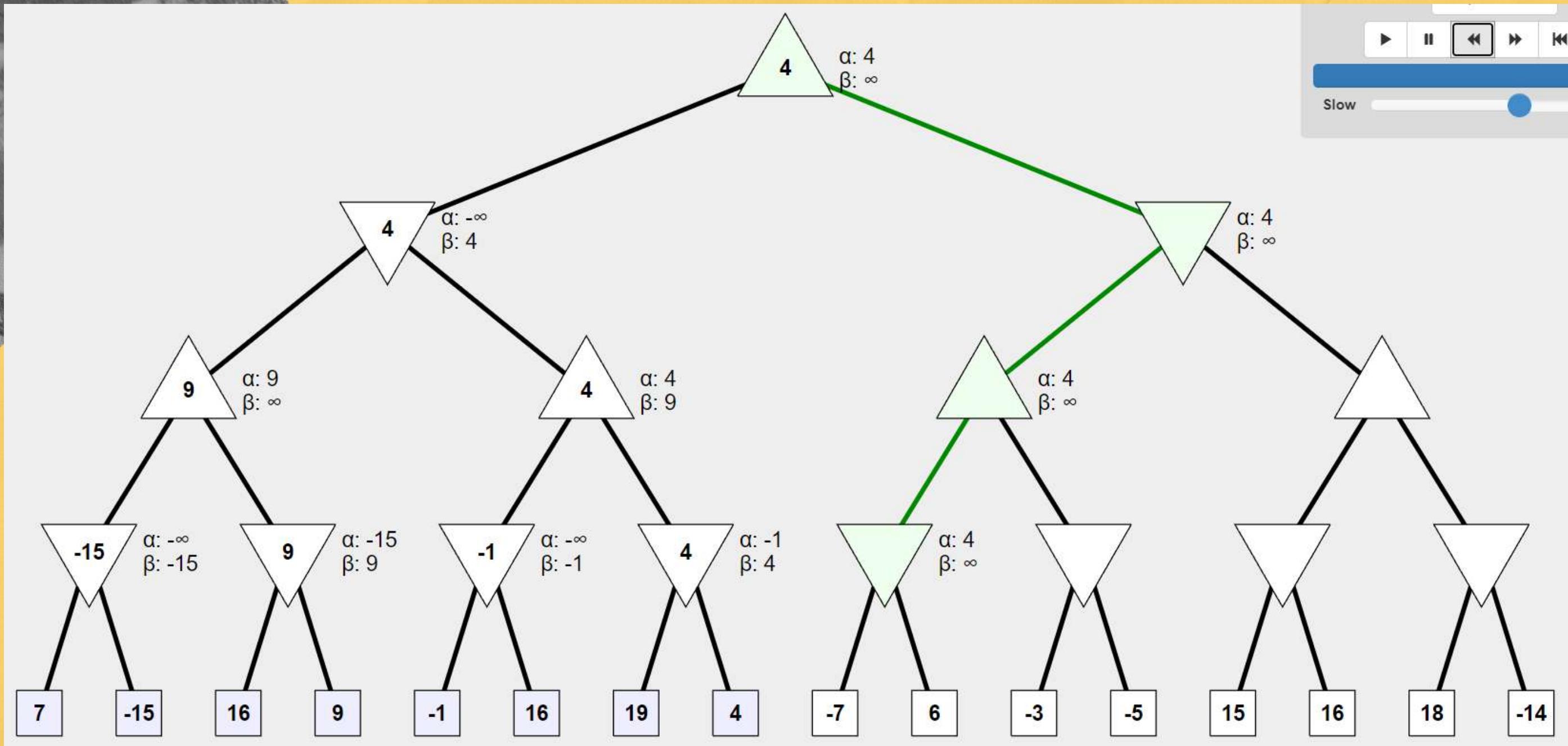




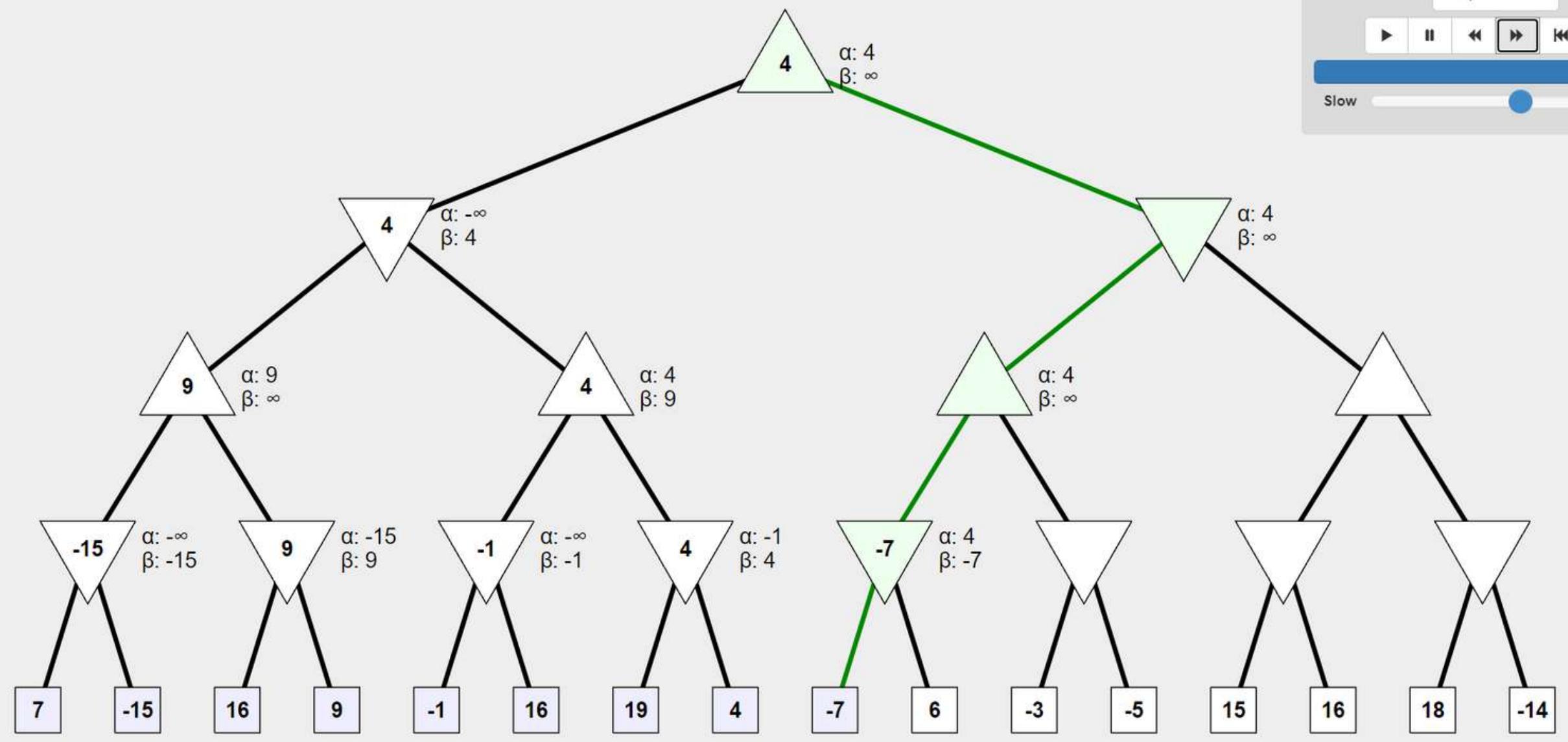
- Kembali ke tingkat di atasnya yaitu move putih.
- Sekarang nilai dari child kanan sudah ditemukan yaitu 4 sehingga memilih nilai maksimum antara 4 dan -1 yaitu 4.
- α diubah menjadi 4 karena $4 > -1$.
- Dilanjutkan ke tingkat di atasnya.



- Kembali ke tingkat di atasnya yaitu move hitam.
- Sekarang nilai dari child kanan sudah ditemukan yaitu 4 sehingga memilih nilai minimum antara 9 dan 4 yaitu 4.
- β diubah menjadi 4 karena $4 < 9$.
- Dilanjutkan ke tingkat di atasnya.

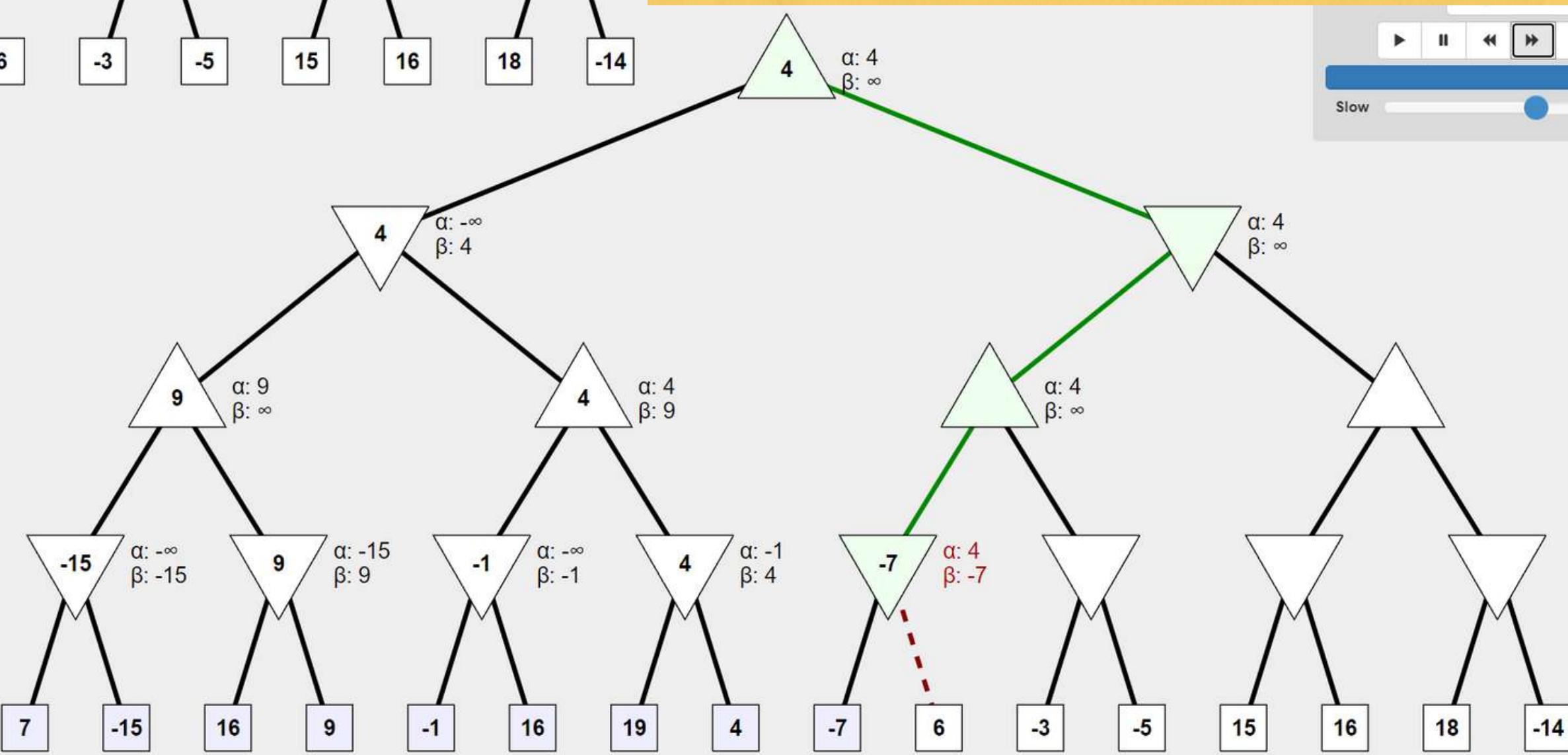


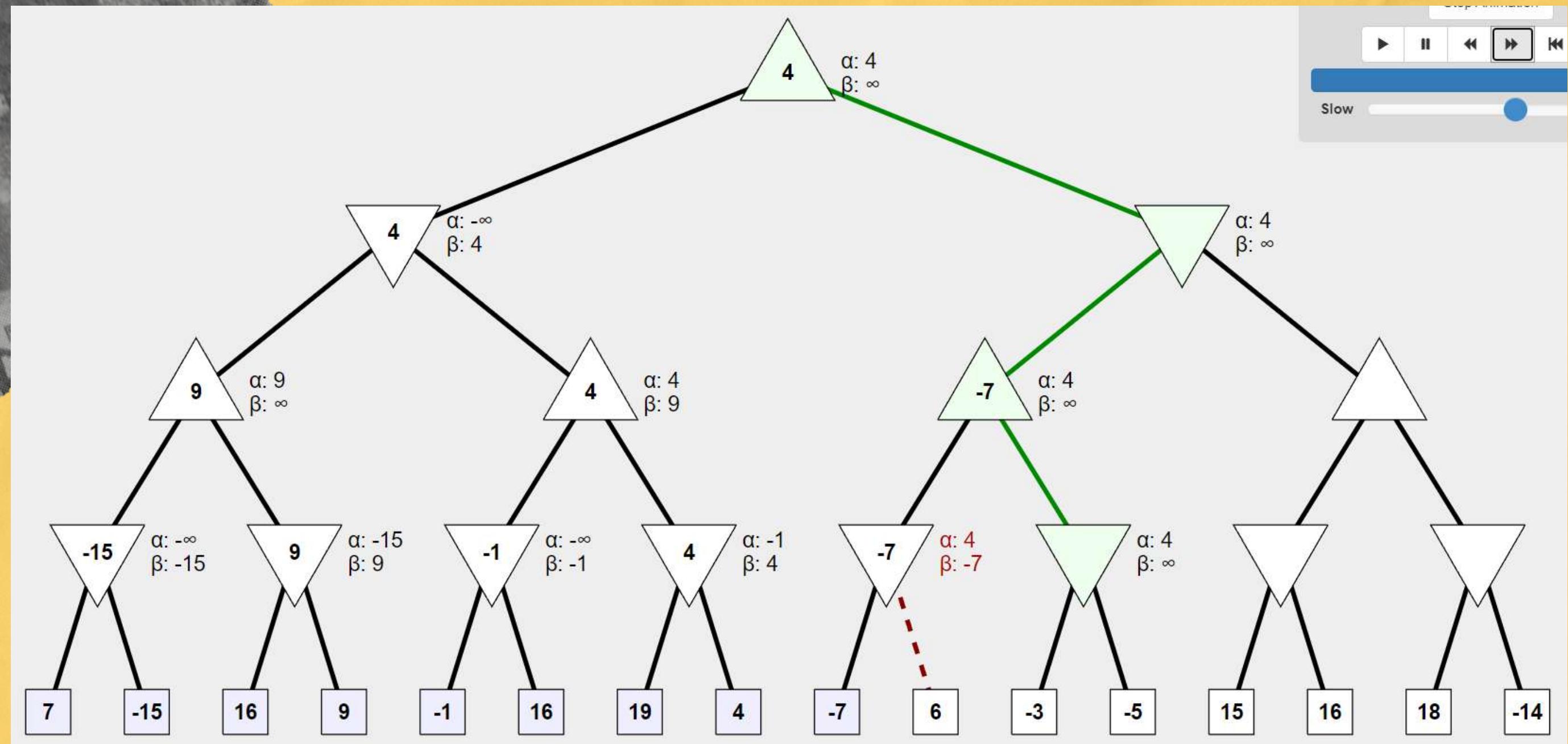
- Dilanjutkan ke tingkat di atasnya yaitu move putih
- Karena nilai yang sudah ditemukan hanya nilai pada child kiri = 4 maka 4 dipilih menjadi nilai maksimum.
- α diubah menjadi 4 karena $4 > -\infty$.
- Dilanjutkan untuk mencari child kanan.



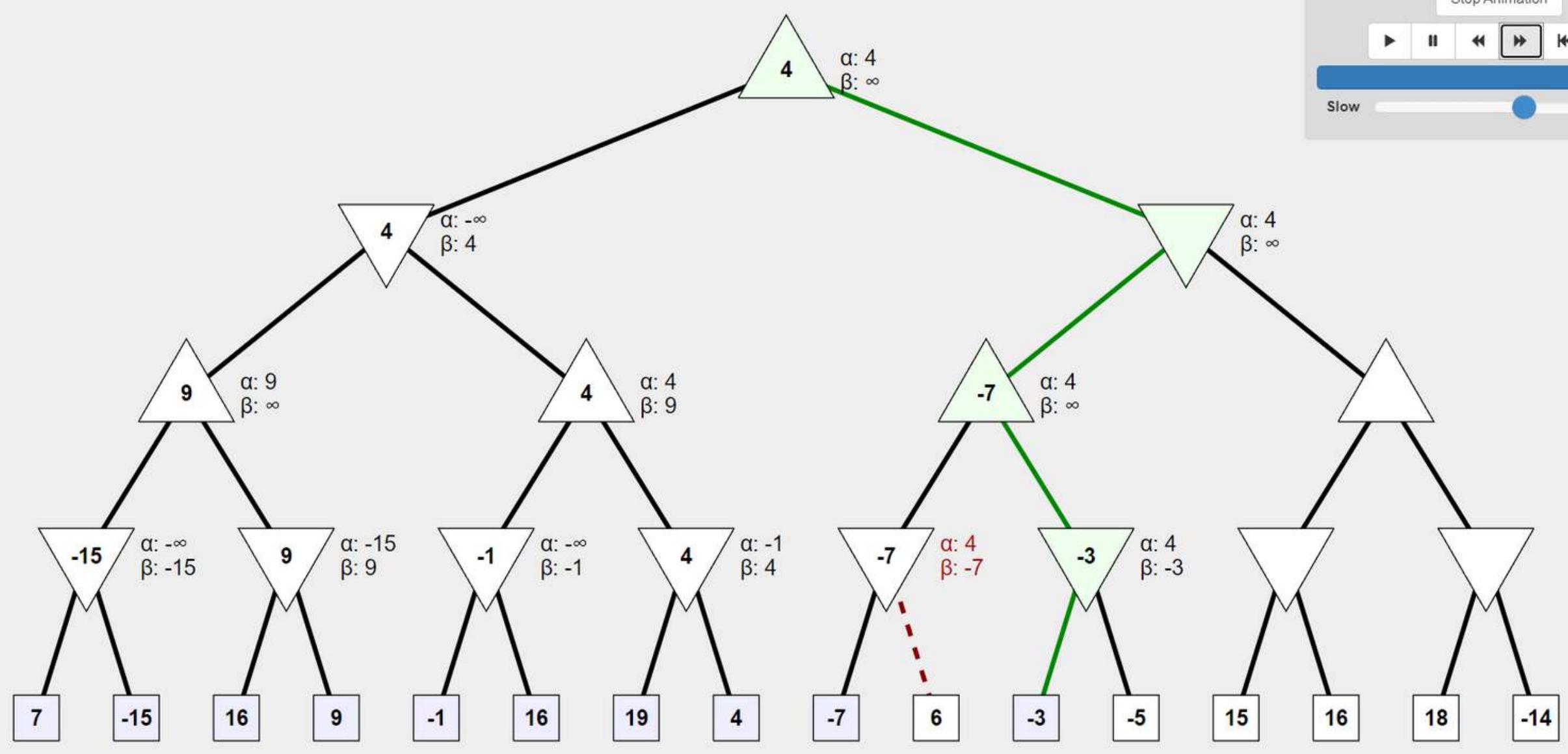
- Dimulai dari leaf kiri.
- Karena nilai dalam hitam masih kosong, maka -7 dipilih secara langsung.
- Lalu -7 dibandingkan dengan nilai β sekarang yaitu ∞ , lalu β diubah menjadi -7 karena $-7 < \infty$.

- Kemudian melanjutkan ke leaf kanan
- Karena $\alpha \geq \beta$ ($4 \geq -7$) maka pencarian pada cabang tersebut dihentikan, karena tidak ada gunanya mengevaluasi cabang-cabang child yang lain.



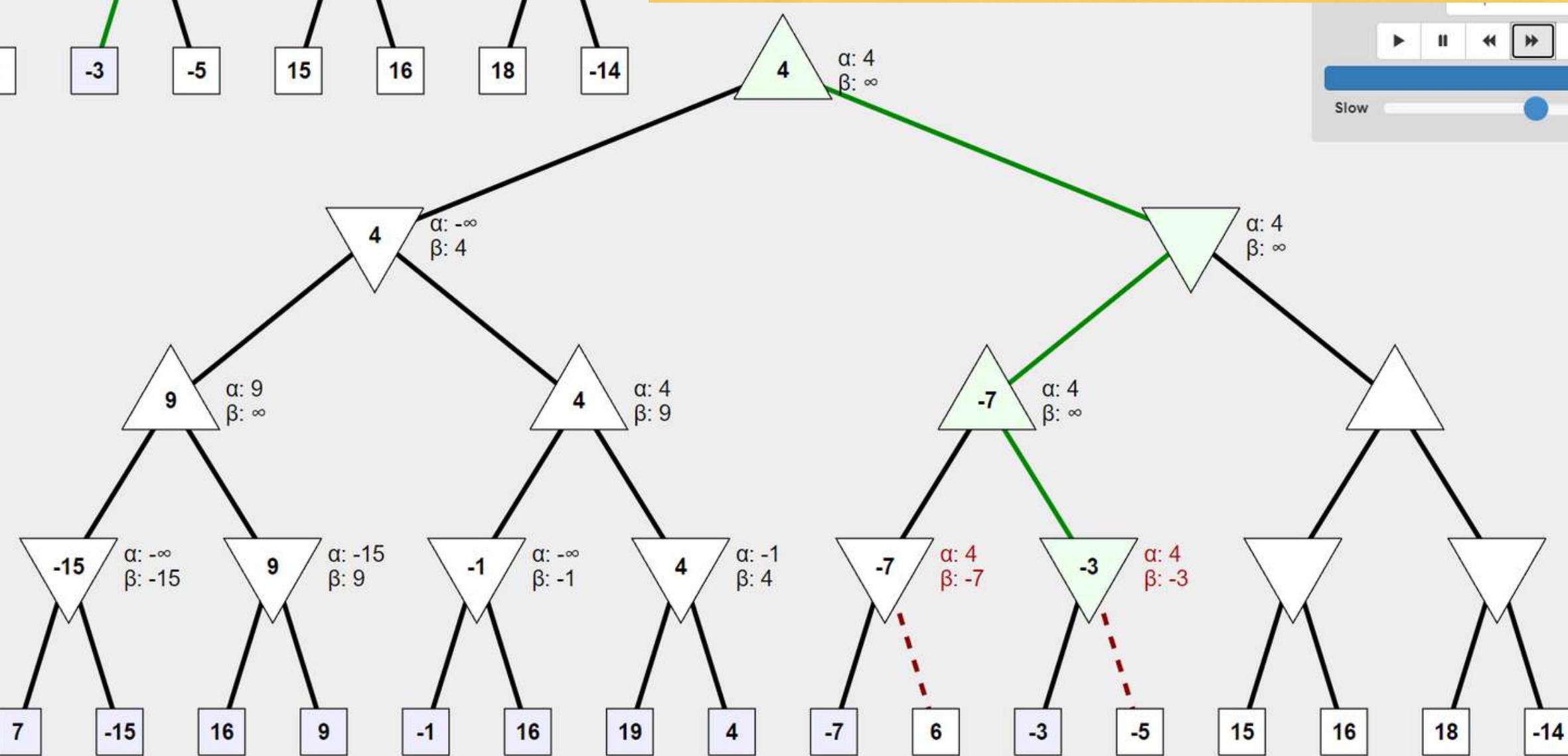


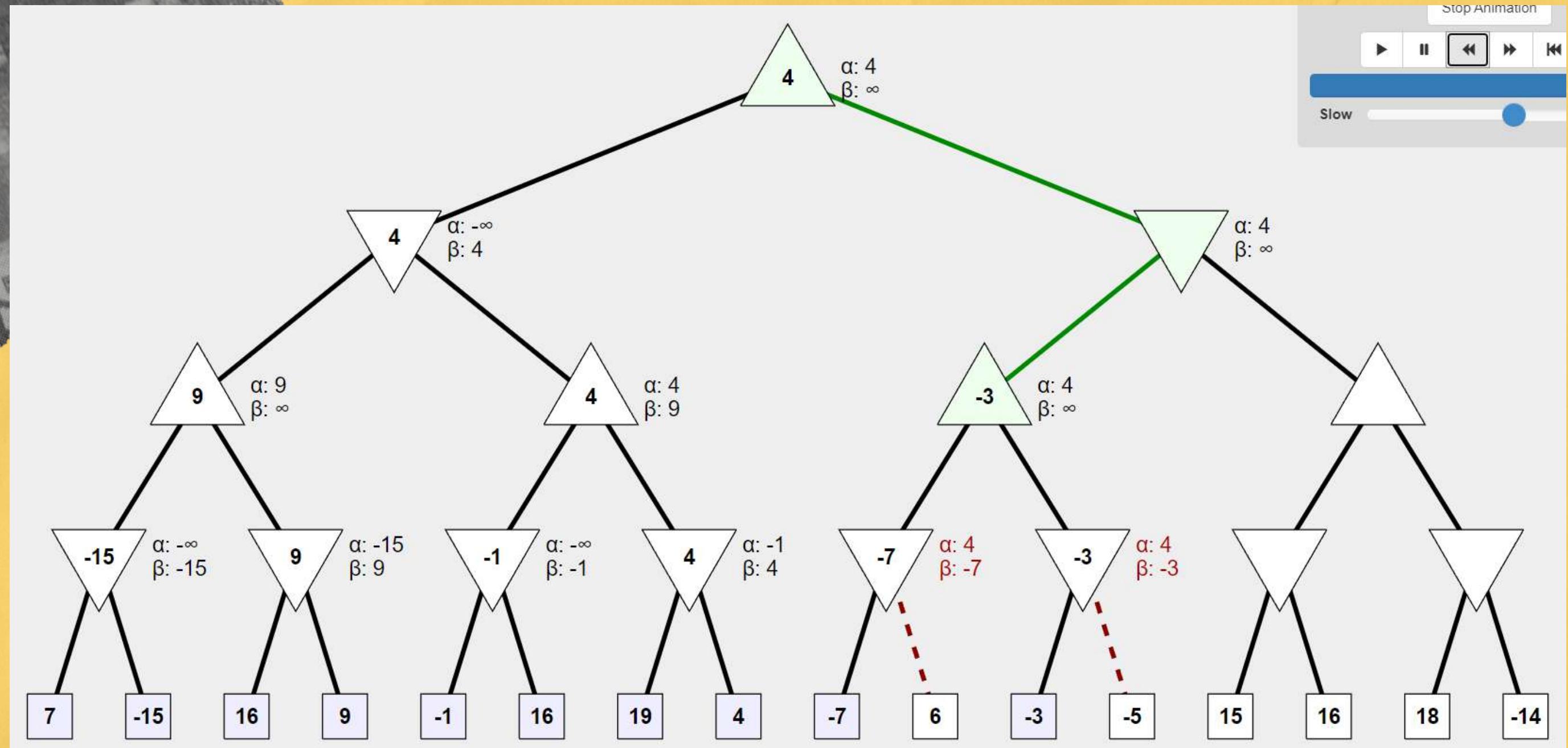
- Dilanjutkan ke tingkat di atasnya yaitu move putih
- Karena nilai yang sudah ditemukan hanya nilai pada child kiri = -7 maka -7 dipilih menjadi nilai maksimum
- α tetap 4 karena $4 > -7$
- Dilanjutkan mencari nilai untuk child kanan



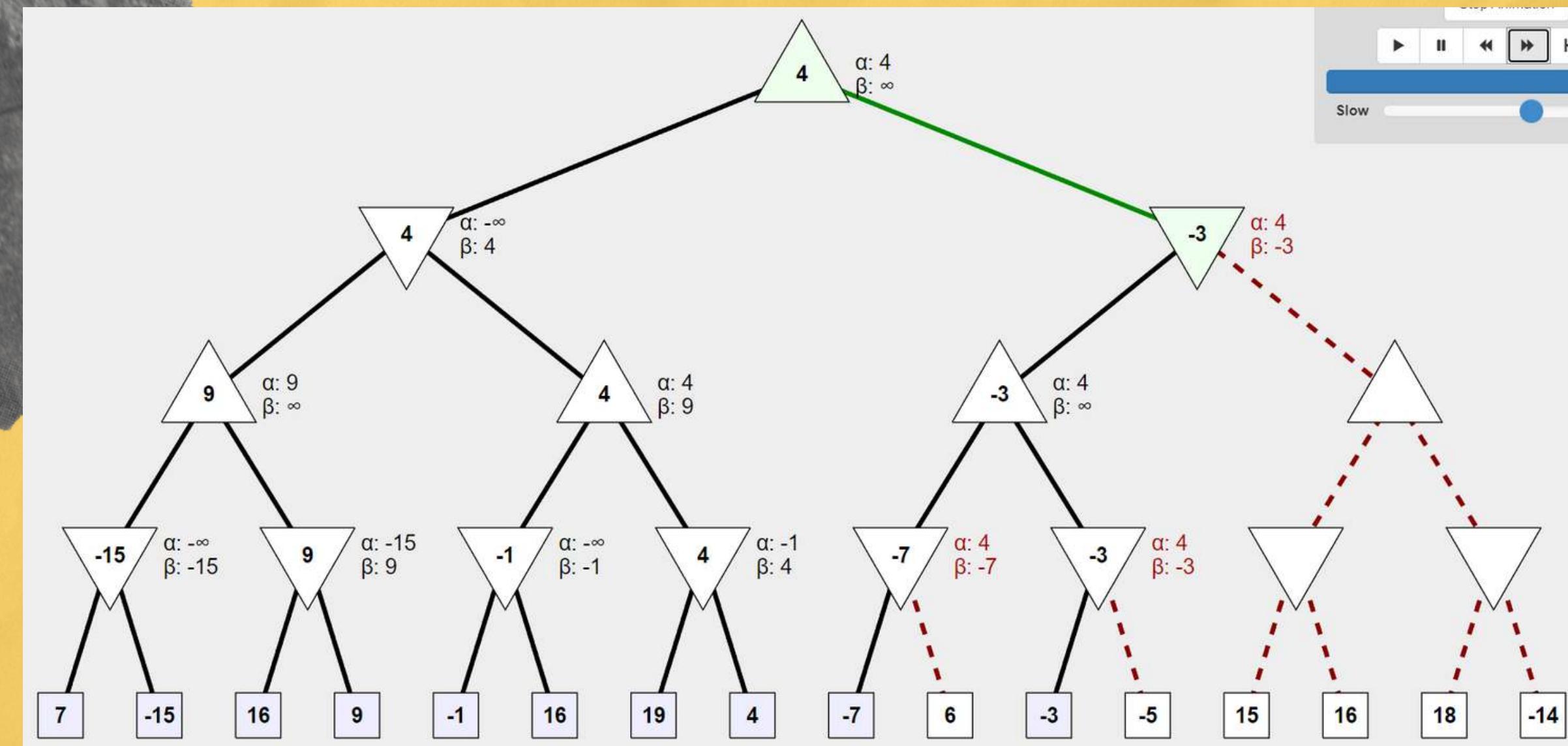
- Dimulai dari leaf kiri.
- Karena nilai dalam hitam masih kosong, maka -3 dipilih secara langsung.
- Lalu -3 dibandingkan dengan nilai β sekarang yaitu ∞ , lalu β diubah menjadi -3 karena $-3 < \infty$.

- Kemudian melanjutkan ke leaf kanan
- Karena $\alpha \geq \beta$ ($4 \geq -3$) maka pencarian pada cabang tersebut dihentikan, karena tidak ada gunanya mengevaluasi cabang-cabang child yang lain.

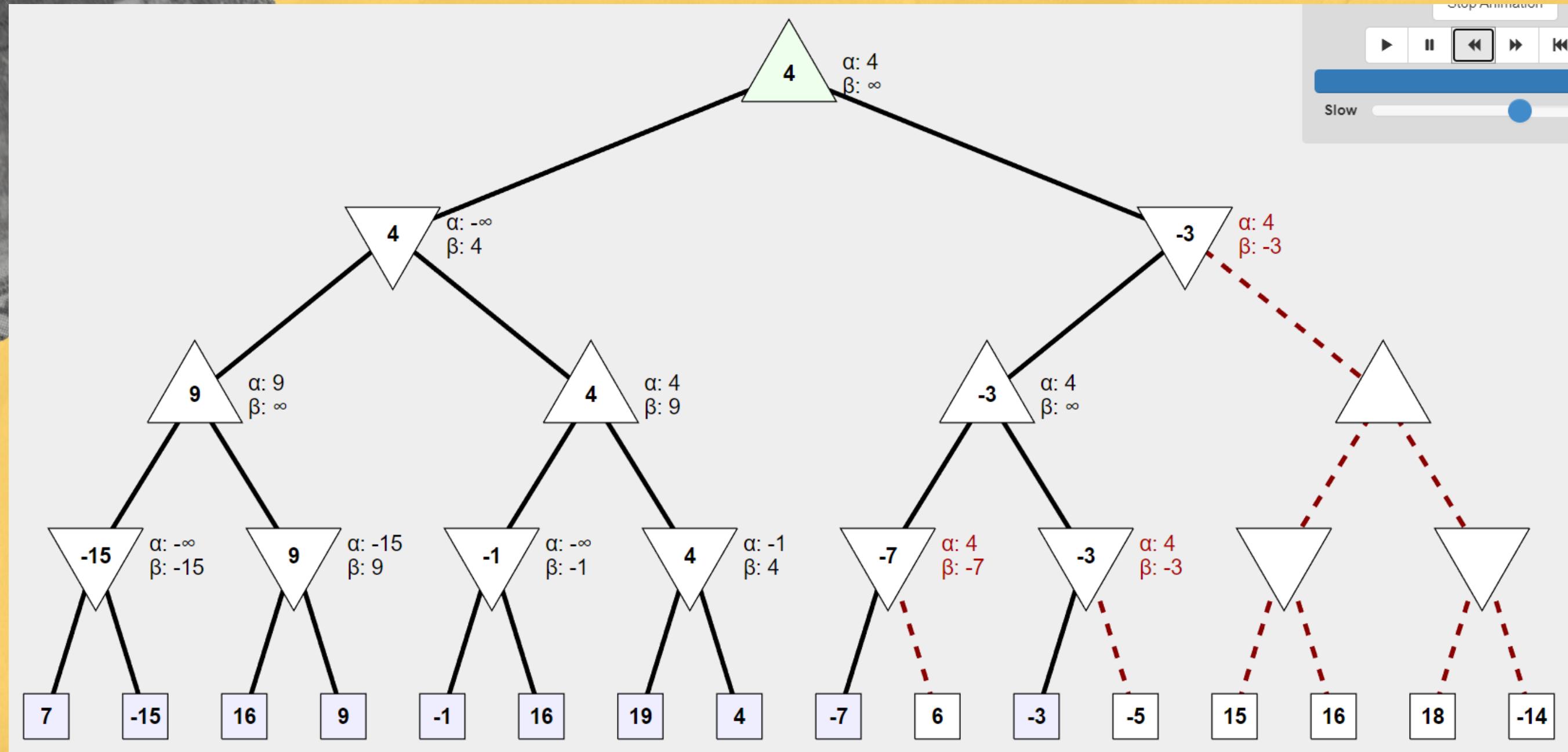




- Kembali ke tingkat di atasnya yaitu move putih.
- Sekarang nilai dari child kanan sudah ditemukan yaitu -3 sehingga memilih nilai maksimum antara -3 dan -7 yaitu -3
- α tetap 4 karena $4 > -3$.
- Dilanjutkan ke tingkat di atasnya.



- Dilanjutkan ke tingkat di atasnya yaitu move hitam.
- Karena nilai yang sudah ditemukan hanya nilai pada child kiri = -3 maka -3 dipilih menjadi nilai minimum
- β diubah menjadi -3 karena $-3 < \infty$.
- Karena hitam selalu mencari nilai yang minimum (paling buruk untuk putih) dan hitam sudah mendapatkan nilai sekurang-kurangnya -3, maka observasi ke subtree kanan dihentikan (prune) karena nilai yang didapat pasti tidak lebih baik dari -3.
- Berlaku $\alpha \geq \beta$ ($4 \geq -3$).



- Lalu ke tingkat di atasnya yaitu move putih.
- Sekarang nilai dari child kanan sudah ditemukan yaitu -3 sehingga memilih nilai maksimum antara 4 dan -3 yaitu 4.
- α tetap 4 karena $4 > -3$.
- Maka hasil akhirnya yaitu 4 (nilai pada root/puncak).



TERIMA KASIH

