

SHORTEST PATH IN GRID 2D

BACKTRACKING

NUR RAMADHANI
D0222302
INFORMATIKA A

APA ITU BACKTRACKING?

Backtracking adalah sebuah teknik algoritma yang digunakan untuk mencari solusi dari masalah dengan mencoba berbagai kemungkinan secara sistematis dan mundur ke langkah sebelumnya jika kemungkinan yang sedang dieksplorasi tidak menghasilkan solusi yang benar atau valid. Teknik ini sering digunakan dalam permasalahan yang melibatkan pengambilan keputusan, pencarian jalur, atau penguraian struktur kombinatorial.

SEJARAH ALGORITMA BACKTRACKING

Algoritma backtracking pertamakali diperkenalkan oleh Edsger W. Dijkstra pada tahun 1960-an dalam konteks pemrograman. Dijkstra adalah seorang ilmuwan komputer terkenal yang memberikan istilah "backtracking" untuk metode ini. Algoritma backtracking digunakan untuk memecahkan masalah kompleks, seperti permainan catur dan masalah jadwal, dengan mencoba semua kemungkinan solusi secara sistematis. Seiring dengan perkembangan komputasi dan pemrograman, algoritma backtracking terus digunakan dan dikembangkan dalam berbagai bidang, dan banyak algoritme penting didasarkan pada konsep backtracking.

Algoritma backtracking telah digunakan sejak awal pengembangan komputasi. Konsep backtracking melibatkan pemecahan masalah dengan mencoba semua kemungkinan solusi secara sistematis, dan jika sebuah solusi tidak memenuhi syarat, maka kembali ke tahap sebelumnya dan mencoba solusi lain.

STRUKTUR DATA YANG DIGUNAKAN

Array dua dimensi Labirin dan Arah digunakan untuk menyimpan struktur labirin dengan menggunakan bilangan bulat sebagai representasi setiap sel dalam labirin dan menyimpan arah pergerakan dari setiap posisi dalam labirin.

Stack digunakan untuk menyimpan posisi saat ini yang sedang dieksplorasi selama pencarian jalur terpendek. Posisi yang baru ditemukan akan ditambahkan ke stack, dan saat posisi tersebut tidak memiliki kemungkinan pergerakan lain, posisi tersebut akan dihapus dari stack.

DESAIN ALGORITMA BACKTRACKING

1. Inisialisasi labirin: Labirin diwakili oleh matriks 'labirin' dengan elemen-elemen berupa angka. Angka -1 mewakili tembok, angka 0 mewakili jalur yang dapat dilalui, dan angka -2 mewakili titik tujuan.
2. Inisialisasi array arah: Array digunakan untuk menyimpan arah yang telah dilalui. Arah 0 adalah utara (north), arah 1 adalah timur (east), arah 2 adalah selatan (south), dan arah 3 adalah barat (west).
3. Inisialisasi stack: Stack digunakan untuk menyimpan posisi saat ini dalam proses pencarian jalur.
4. Inisialisasi posisi awal: Posisi awal ditentukan oleh 'startX' dan 'startY' dan disimpan dalam objek 'start'. Posisi awal dimasukkan ke dalam stack.

DESAIN ALGORITMA BACKTRACKING

5. Melakukan pencarian jalur dengan menggunakan algoritma backtracking:
 - Selama stack tidak kosong, ambil posisi teratas dari stack.
 - Periksa arah saat ini, nilai labirin pada posisi tersebut, dan nilai labirin pada posisi berikutnya.
 - Tentukan posisi depan, kanan, dan kiri berdasarkan arah saat ini.
 - Jika posisi depan, kanan, atau kiri merupakan jalur yang dapat dilalui, perbarui nilai labirin, arah, dan masukkan posisi tersebut ke dalam stack.
 - Jika posisi depan, kanan, atau kiri merupakan titik tujuan (-2), cetak jumlah langkah dan kata "FINISH", dan berhenti.
 - Jika tidak ada jalur yang dapat dilalui, hapus posisi saat ini dari stack (operasi pop).
6. Cetak stack setiap langkahnya untuk melihat jalur yang sedang dieksplorasi.
7. Cetak labirin setelah pencarian selesai untuk melihat jalur terpendek yang ditemukan.



TERIMA KASIH

reallygreatsite.com
