



ITS

Institut
Teknologi
Sepuluh Nopember

PRESENTASI TUGAS AKHIR - KI141502

DESAIN DAN ANALISIS ALGORITMA DIJKSTRA DAN METODE VISIBILITY GRAPH NAIVE UNTUK PENYELESAIAN persoalan SPOJ 780 THE ARCHIPELAGO

Penyusun Tugas Akhir:

Reva Yoga Pradana
(5112 100 062)

Dosen Pembimbing:

Victor Hariadi, S.Si., M.Kom.
Rully Soelaiman, S.Kom., M.Kom.



ITS

Institut
Teknologi
Sepuluh Nopember

PENDAHULUAN

TINJAUAN PUSTAKA

STRATEGI PENYELESAIAN

UJI COBA

KESIMPULAN



ITS

Institut
Teknologi
Sepuluh Nopember

Latar Belakang

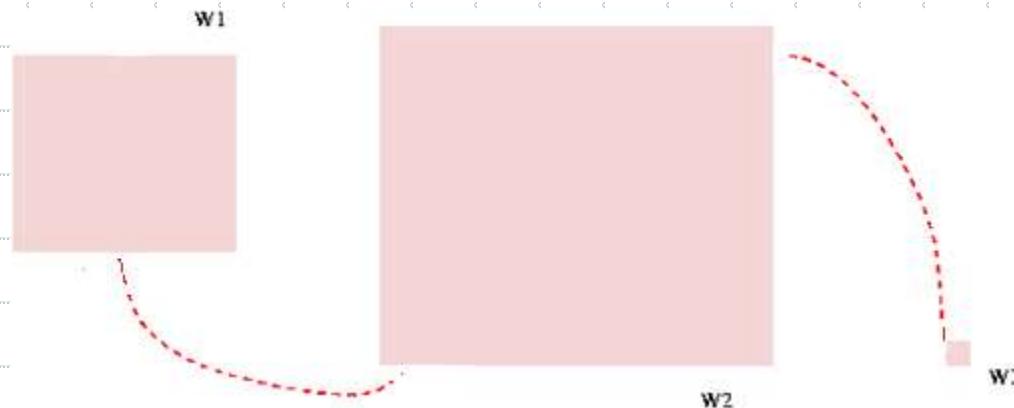
Rumusan Masalah

Tujuan

Batasan Masalah

LATAR BELAKANG

- Permasalahan yang diangkat dalam tugas akhir ini diambil dari persoalan SPOJ dengan nomor soal 780, kode soal ARCHPLG dan berjudul “*The Archipelago*”.
- Pada permasalahan *The Archipelago*, terdapat sebuah negara fiktif bernama *Byteland*.
- *Byteland* memiliki beberapa pulau berbentuk persegi panjang. Beberapa pulau dapat terhubung oleh jalur kapal ferry.





ITS

Institut
Teknologi
Sepuluh Nopember

Latar Belakang

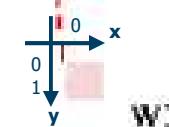
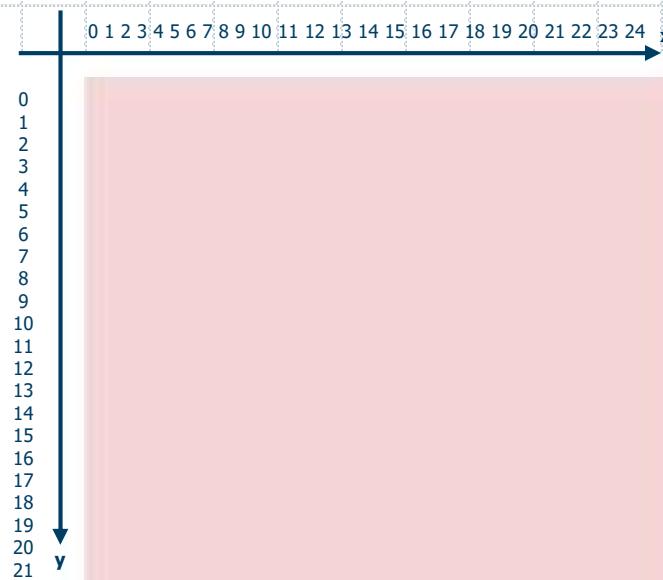
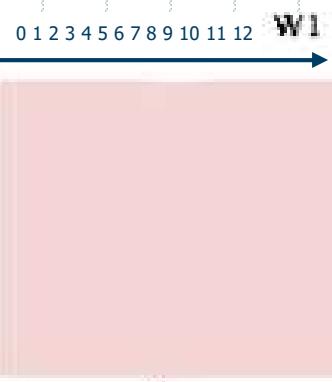
Rumusan Masalah

Tujuan

Batasan Masalah

LATAR BELAKANG

- Setiap pulau digambarkan dalam bidang geometri, yang mana memiliki sistem koordinat tersendiri di tiap pulaunya.





ITS

Institut
Teknologi
Sepuluh Nopember

Latar Belakang

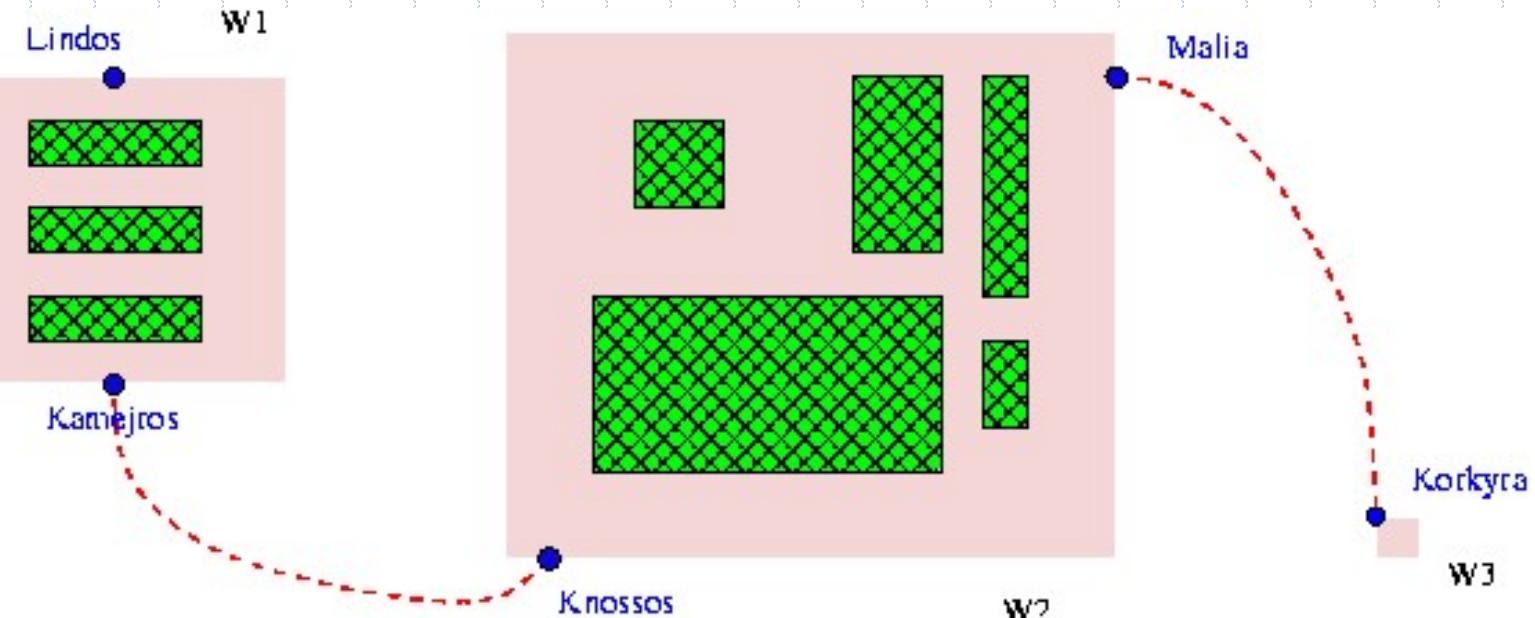
Rumusan Masalah

Tujuan

Batasan Masalah

LATAR BELAKANG

- Dalam setiap pulau, terdapat 2 elemen, yaitu: area terlarang (berbentuk persegi panjang) dan terminal.
- Area terlarang tidak boleh dilewati (dalam pengambilan rute terpendek)
- Terminal merupakan pelabuhan dari Kapal Ferry.





ITS

Institut
Teknologi
Sepuluh Nopember

Latar
Belakang

Rumusan
Masalah

Tujuan

Batasan Masalah

RUMUSAN MASALAH

- Bagaimana menyelesaikan permasalahan “The Archipelago” pada situs penilaian SPOJ?
- Bagaimana kinerja penyelesaian permasalahan “The Archipelago”?

TUJUAN

- Melakukan desain serta analisis penyelesaikan permasalahan “The Archipelago” pada situs penilaian SPOJ.
- Melakukan analisis terhadap kinerja penyelesaian permasalahan “The Archipelago”.



ITS

Institut
Teknologi
Sepuluh NopemberLatar
BelakangRumusan
Masalah

Tujuan

Batasan Masalah

BATASAN MASALAH

- Permasalahan dalam tugas akhir ini adalah permasalahan “*The Archipelago*” pada situs penilaian SPOJ dengan nomor soal 780 dan kode soal ARCHPLG.
- Implementasi dilakukan dengan bahasa pemrograman C++.
- Terdapat T (jumlah kasus uji)
- Batasan dalam setiap kasus uji:
 - $1 \leq N_{\text{Island}} \leq 1000$
 - $0 \leq x, y \leq 250$
 - $0 \leq N_{\text{Terminal}} \leq 10$
 - $0 \leq N_{\text{Restricted Area}} < 20$
 - $0 \leq m \leq 100000$
- Waktu eksekusi maksimal 5.393 detik
- Penggunaan memori maksimal 1536 MB
- Batas sumber kode 50000 B



ITS

Institut
Teknologi
Sepuluh Nopember

PENDAHULUAN

TINJAUAN PUSTAKA

STRATEGI PENYELESAIAN

UJI COBA

KESIMPULAN



ITS

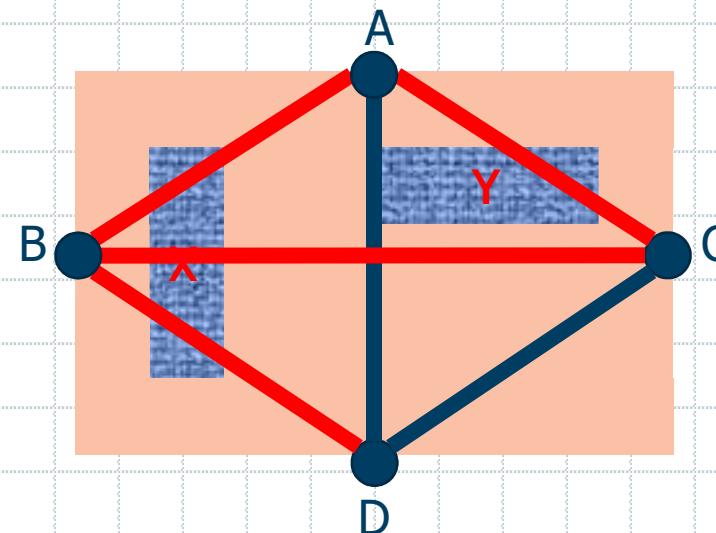
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

VISIBILITY GRAPH

- Visibility Graph digunakan untuk menentukan apakah suatu *vertex visible* / dapat dijangkau oleh *vertex* yang lain.
- Apabila 2 buah *vertex* saling *visible*, maka dapat dibentuk *edge* antar keduanya.



Visibilitas tiap Vertex terhadap Vertex Lain



ITS

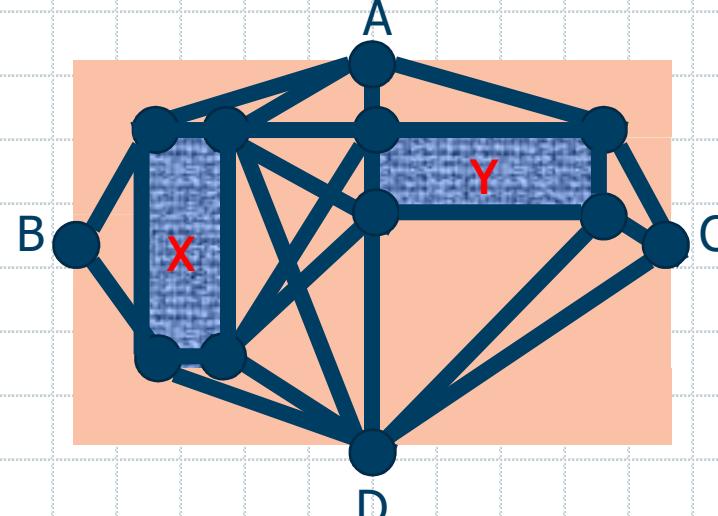
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

PRINSIP DASAR VISIBILITY GRAPH

- Menganggap semua *interest point* menjadi sebuah *vertex*.
- Perlu pengecekan antar *vertex* dapat terbentuk suatu *edge* atau tidak.
- Mempermudah penentuan rute antar *vertex*, terutama untuk *vertex* yang saling tidak *visible*.



Setelah Pemanggilan *Visibility Graph*



ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

VISIBILITY GRAPH NAIVE

- Salah satu algoritma yang dapat menyelesaikan visibility graph, dengan kompleksitas waktu sebesar $O(N^3)$.
- Algoritma Visibility Graph Naive menentukan apakah suatu vertex dapat dijangkau dari vertex yang lain, dengan cara mencoba semua kemungkinan yang ada (mengeceknya dengan semua halangan).



ITS

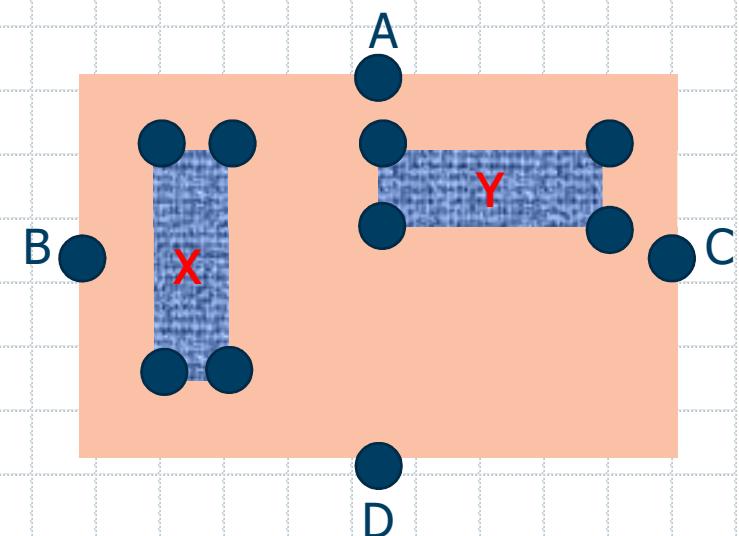
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

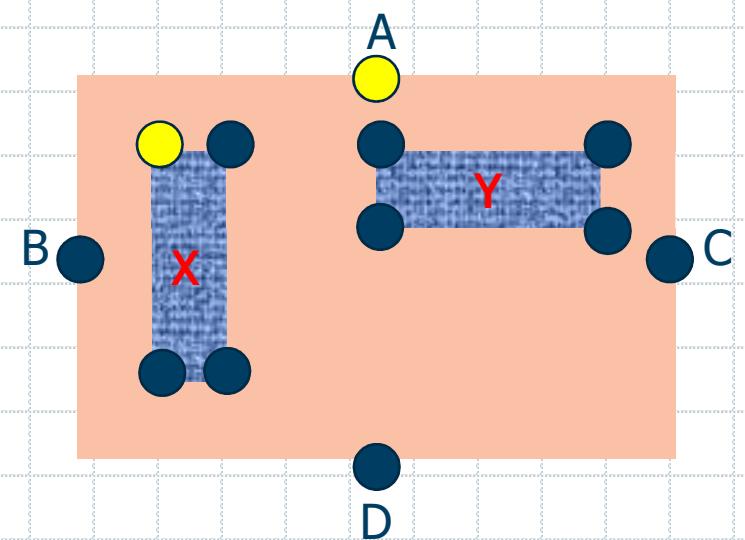
VISIBILITY GRAPH NAIVE

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru



VISIBILITY GRAPH NAIVE

- Berikut langkah-langkah dari algoritma *visibility graph* naive:
 - Tiap vertex cek dengan vertex yang lain
 - Bentuk garis lurus antara kedua vertex tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru





ITS

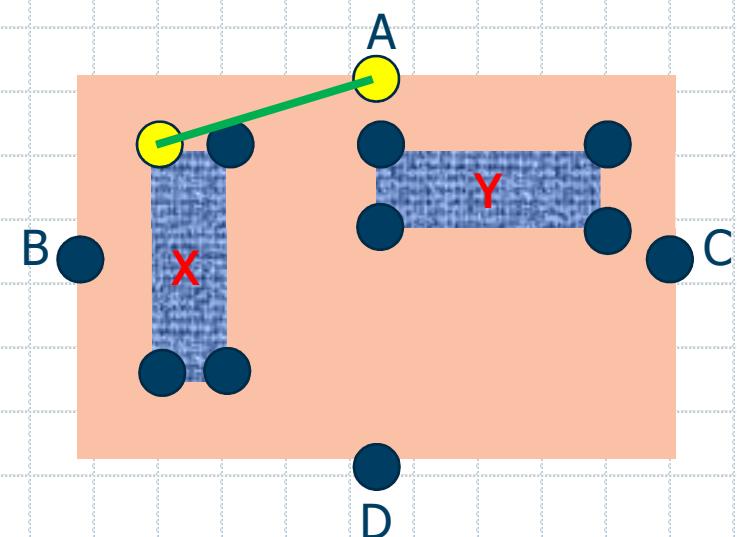
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

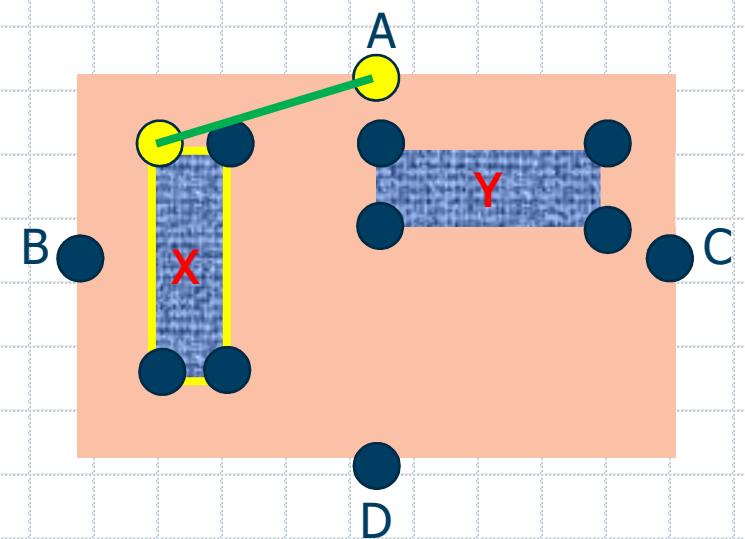
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

VISIBILITY GRAPH NAIVE

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru





ITS

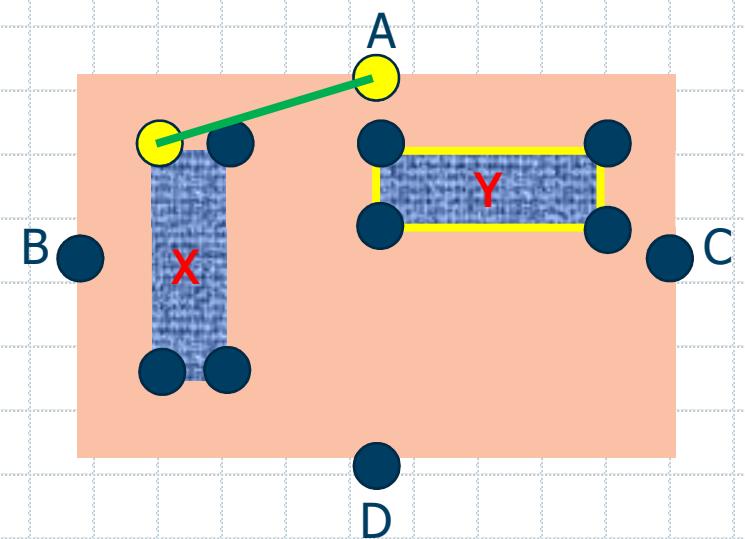
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

VISIBILITY GRAPH NAIVE

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru





ITS

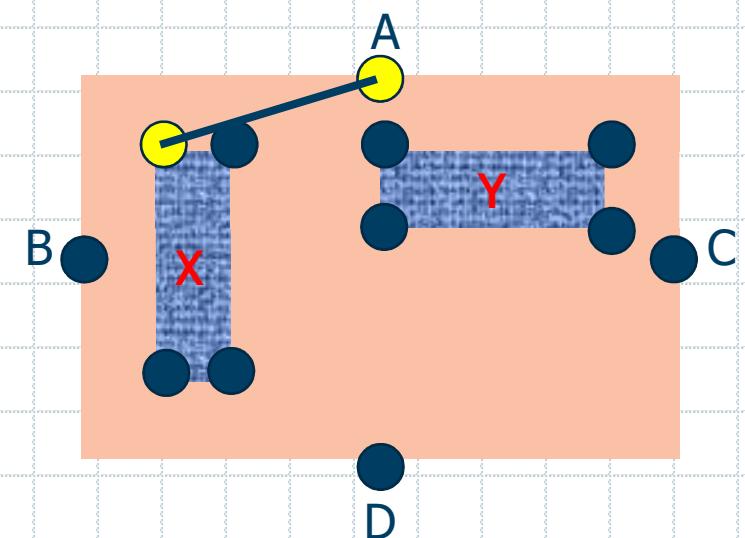
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

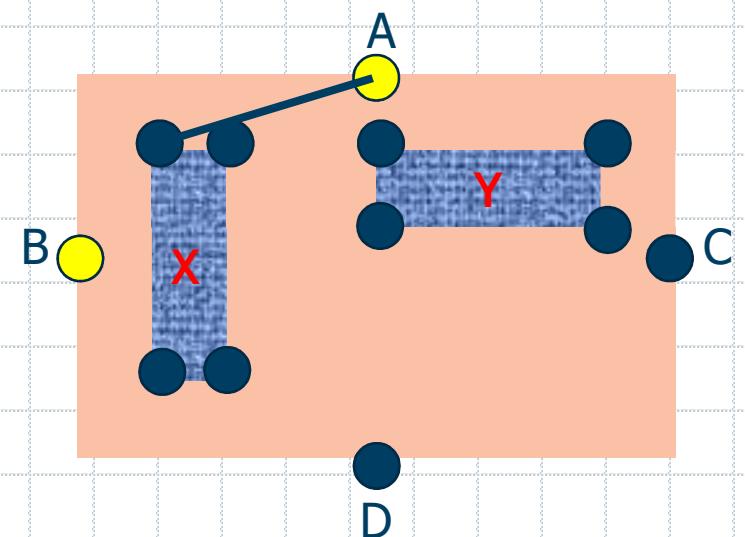
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap vertex cek dengan vertex yang lain
 - Bentuk garis lurus antara kedua vertex tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

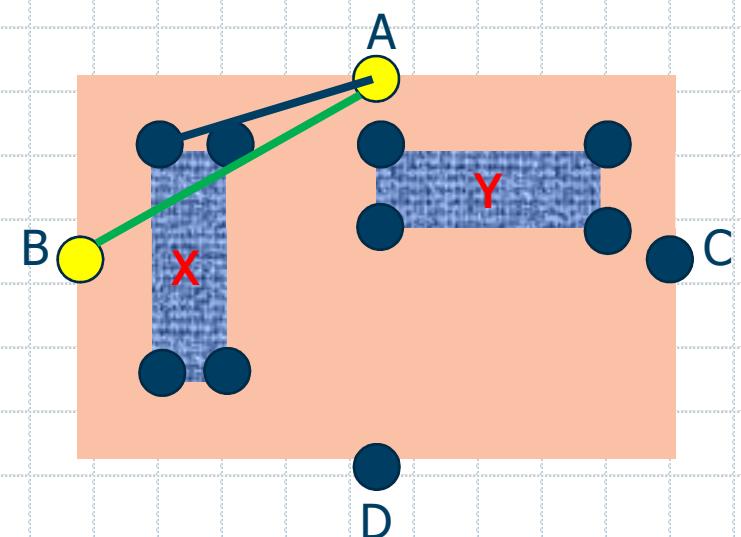
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

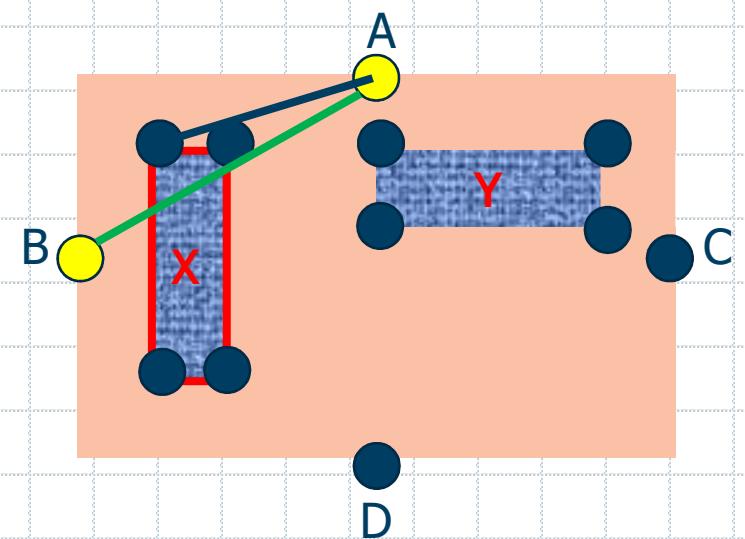
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

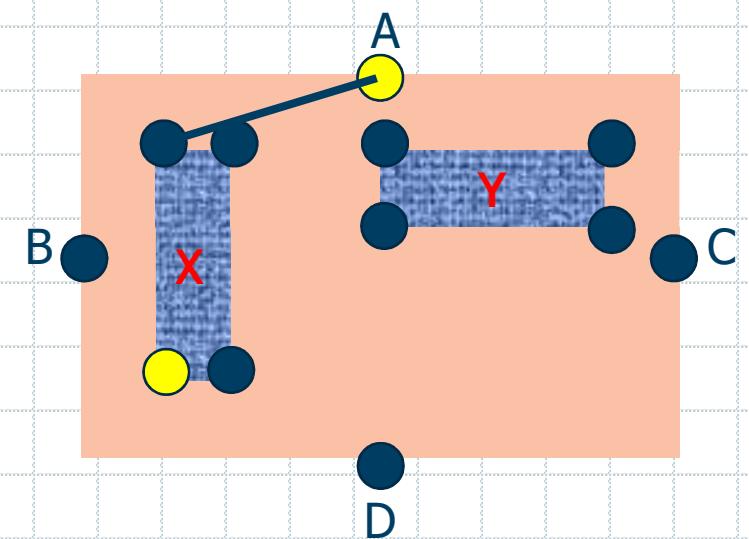
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

VISIBILITY GRAPH NAIVE

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap vertex cek dengan vertex yang lain
 - Bentuk garis lurus antara kedua vertex tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru





ITS

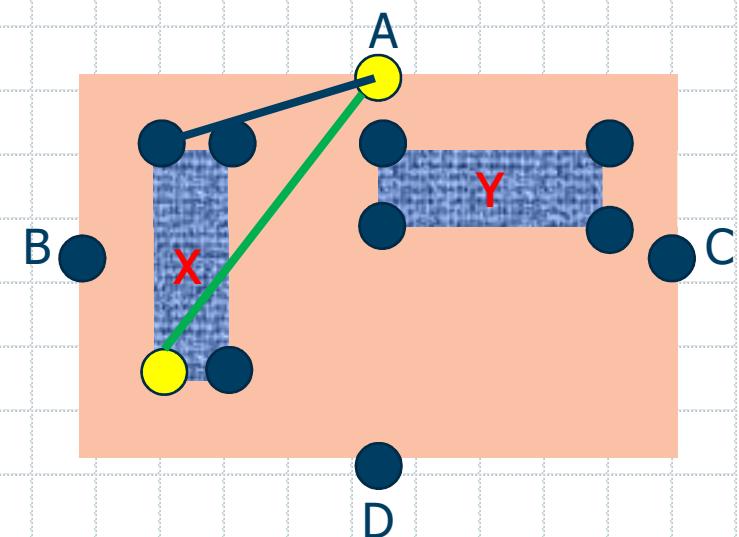
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

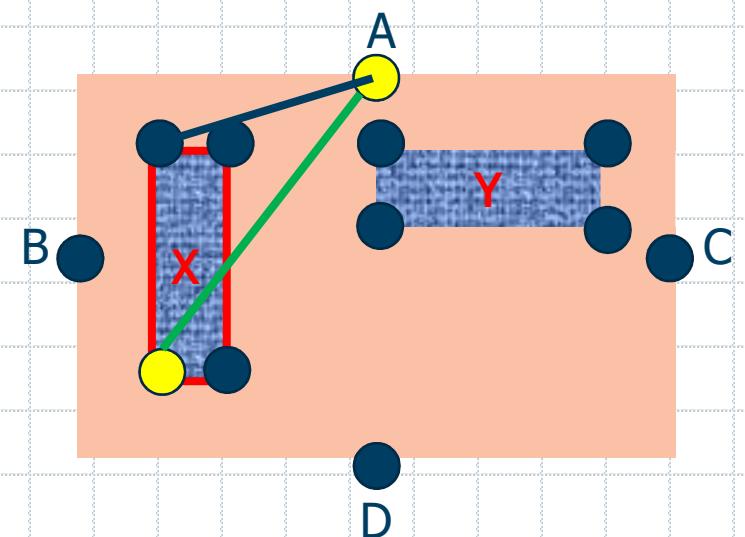
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

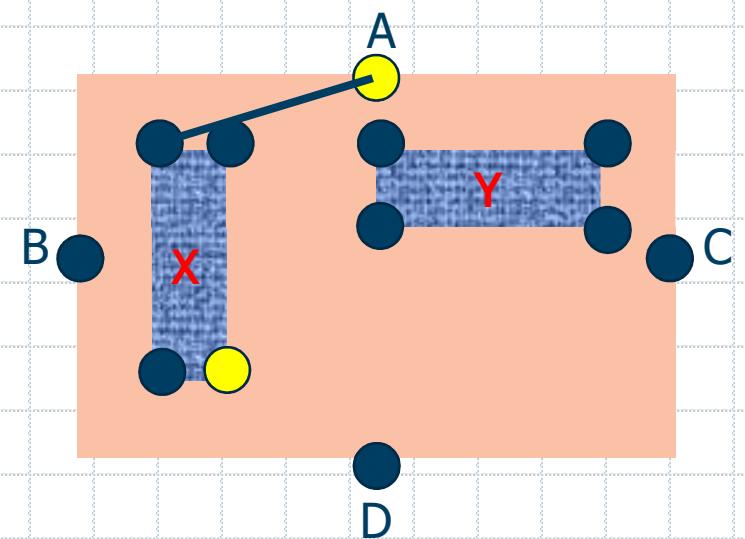
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap vertex cek dengan vertex yang lain
 - Bentuk garis lurus antara kedua vertex tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

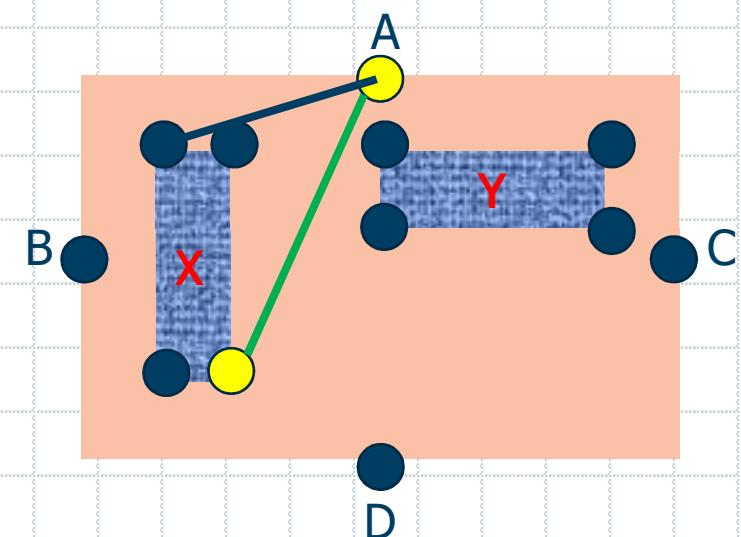
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

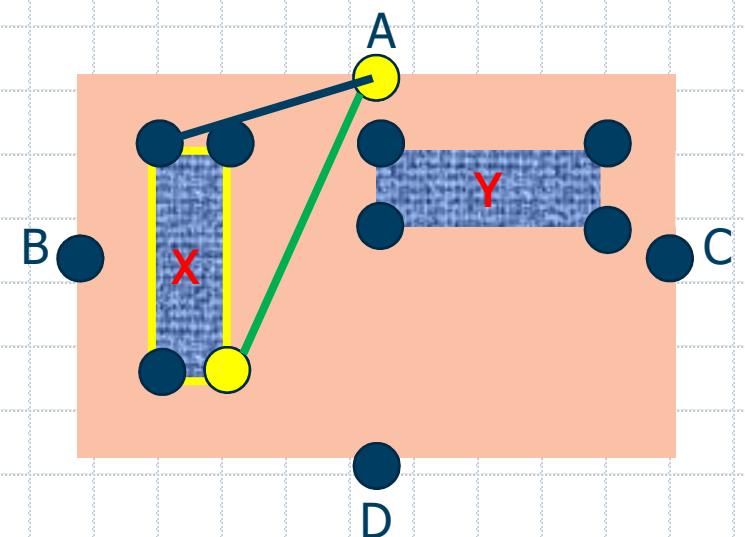
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

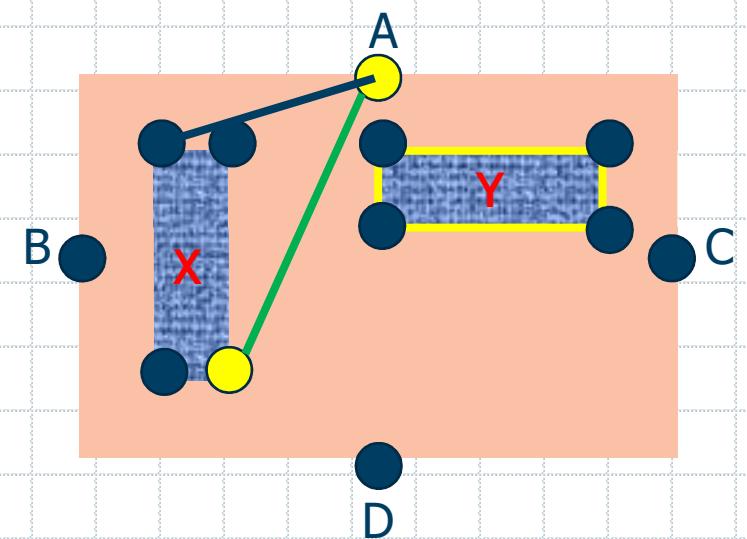
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

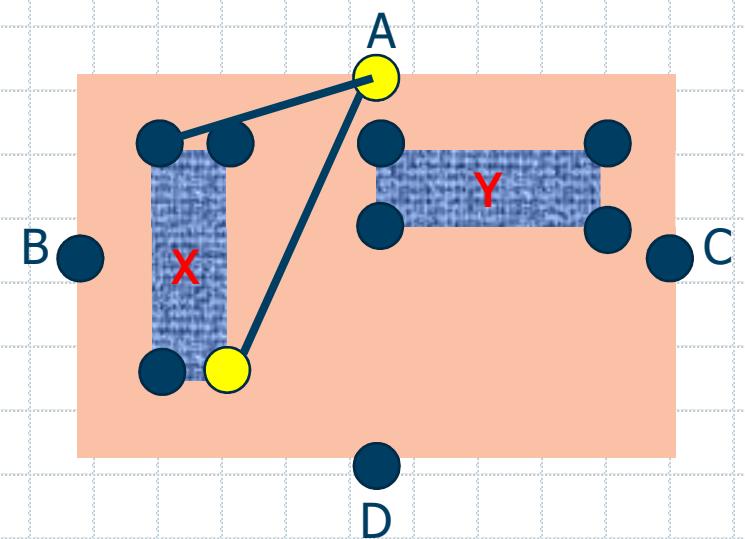
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

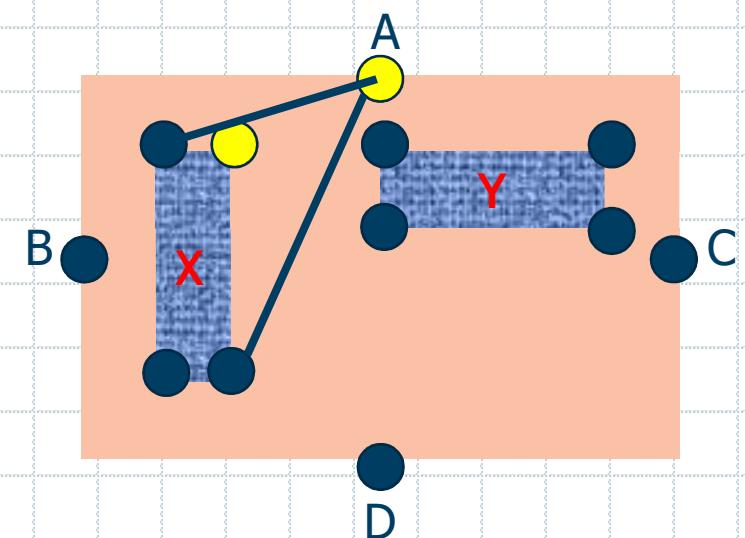
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap vertex cek dengan vertex yang lain
 - Bentuk garis lurus antara kedua vertex tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

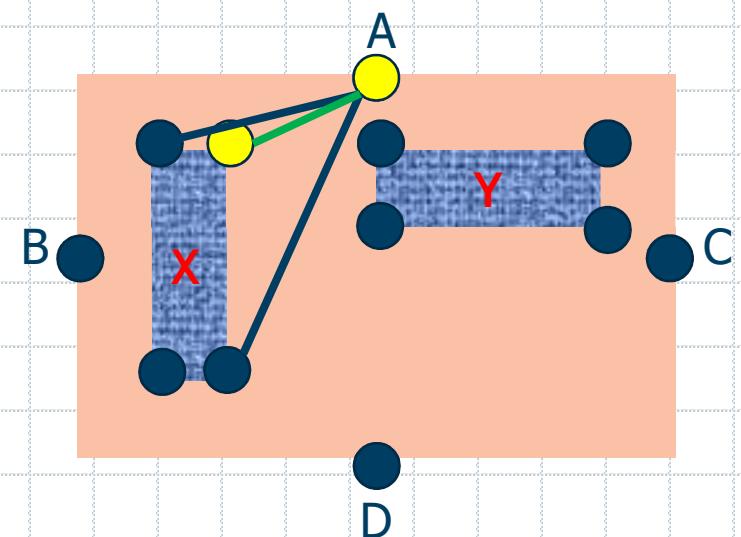
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

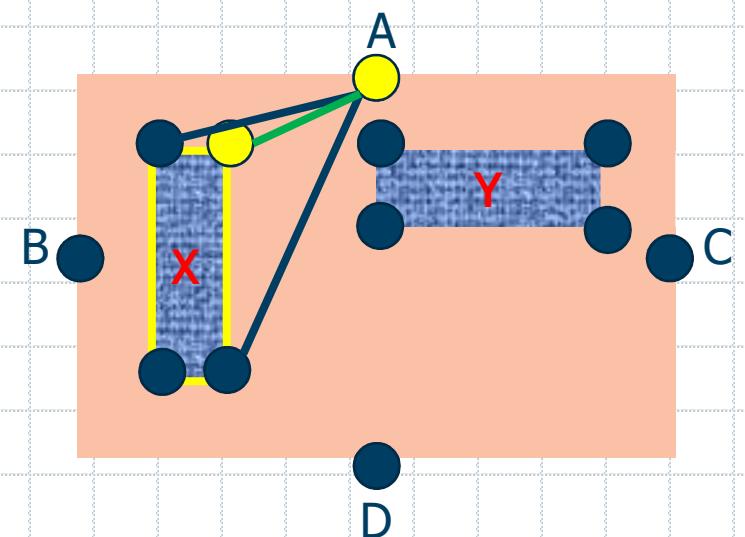
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

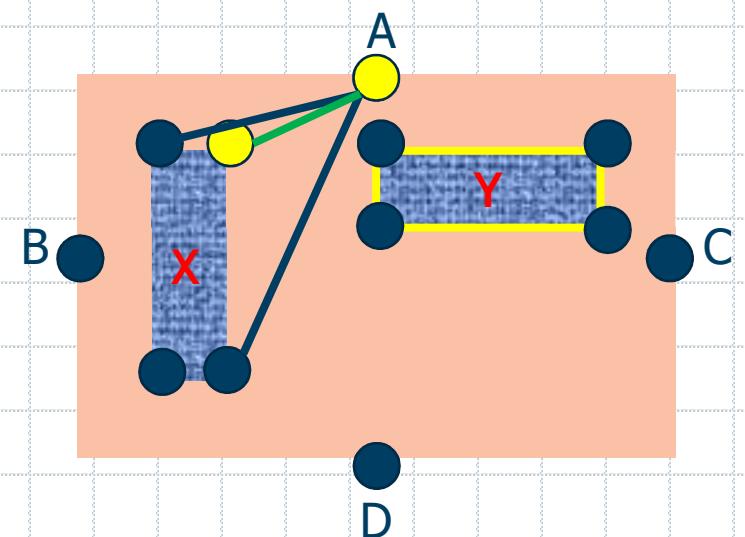
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

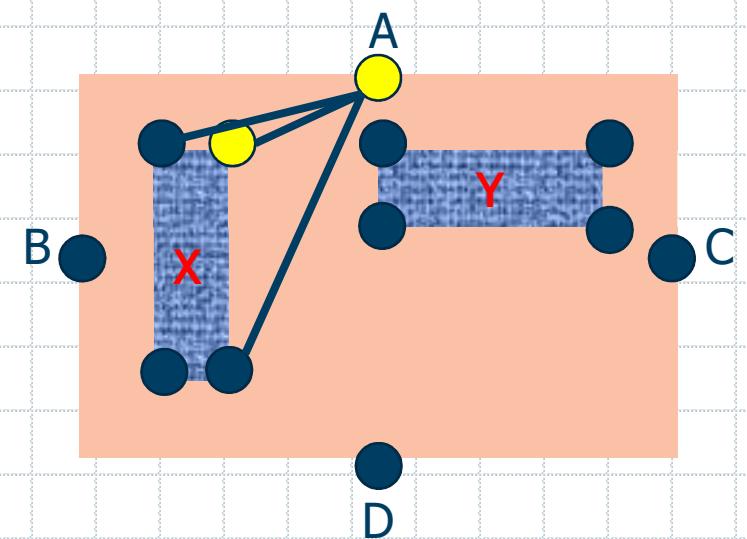
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

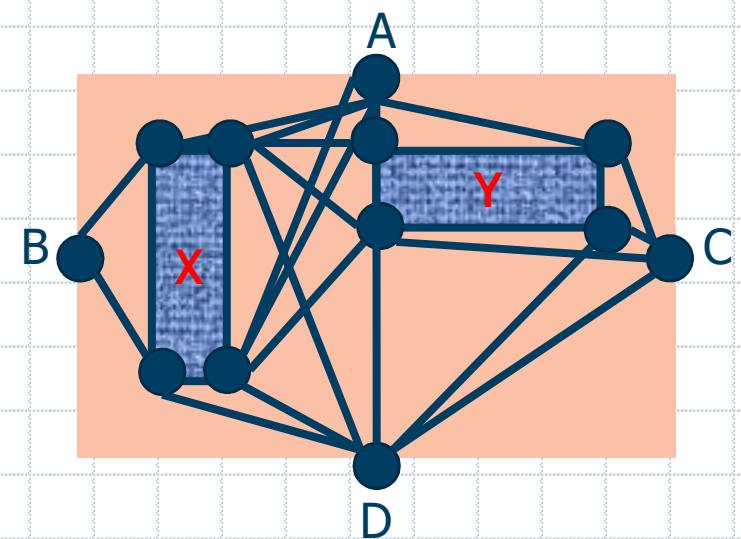
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Berikut langkah-langkah dari algoritma *visibility graph naive*:
 - Tiap *vertex* cek dengan *vertex* yang lain
 - Bentuk garis lurus antara kedua *vertex* tersebut.
 - Tiap garis tersebut cek dengan tiap halangan yang ada.
 - Cek apakah garis lurus tadi bertabrakan dengan salah-satu halangan
 - ◆ Tidak menabrak : buat edge baru

VISIBILITY GRAPH NAIVE





ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

SPOJ THE ARCHIPELAGO

- Diberi masukan T (jumlah kasus uji)
- Untuk tiap kasus uji, diberi masukan berupa:
 - N_{Island} dan m .
 - Tiap N_{Island} terdapat masukan berupa $N_{Terminal}$, $N_{Restricted_Area}$
 - ◆ Tiap $N_{Terminal}$ terdapat masukan $name_{terminal}$, x dan y
 - ◆ Tiap $N_{Restricted_Area}$ terdapat masukan x_1 , y_1 , x_2 , y_2
 - (x_1, y_1) : koordinat titik pojok kiri atas area terlarang
 - (x_2, y_2) : koordinat titik pojok kanan bawah area terlarang
 - Tiap m terdapat masukan $name1$, $island1$, $name2$, $island2$, w
 - Setelah itu, terdapat masukan soal: $name_1$, $island_1$, $name_2$, $island_2$, w



ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

SPOJ THE ARCHIPELAGO

- Ketentuan rute terpendek:

- Rute terpendek bisa melewati jalur kapal ferry dan jalur darat di pulau tertentu.
- Rute pada jalur darat tidak boleh melewati area terlarang.
- Total jarak pada jalur darat harus dibulatkan ke atas (*ceiling*).



ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

case t Y
D
position[0]
position[1]
...
position[last]

SPOJ THE ARCHIPELAGO

Catatan:

- $t \rightarrow$ nomor kasus uji
- $D \rightarrow$ total jarak
- Position:
 - Jika position = terminal
 - cetak name_terminal dan name_island
 - Jika tidak, cetak koordinatnya



ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

Prinsip Dasar

Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

ILUSTRASI PERSOALAN

- Terdapat 3 pulau: W1, W2, W3



ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

Prinsip Dasar

Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

- Pulau W1:

ILUSTRASI PERSOALAN



ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

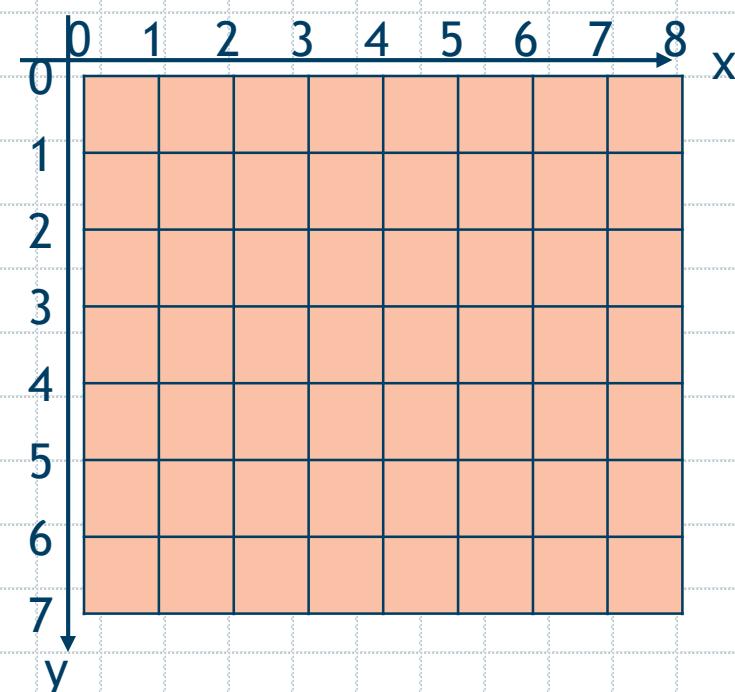
Prinsip Dasar

Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

- Pulau W1:
 - Ukuran 8x7





ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

Prinsip Dasar

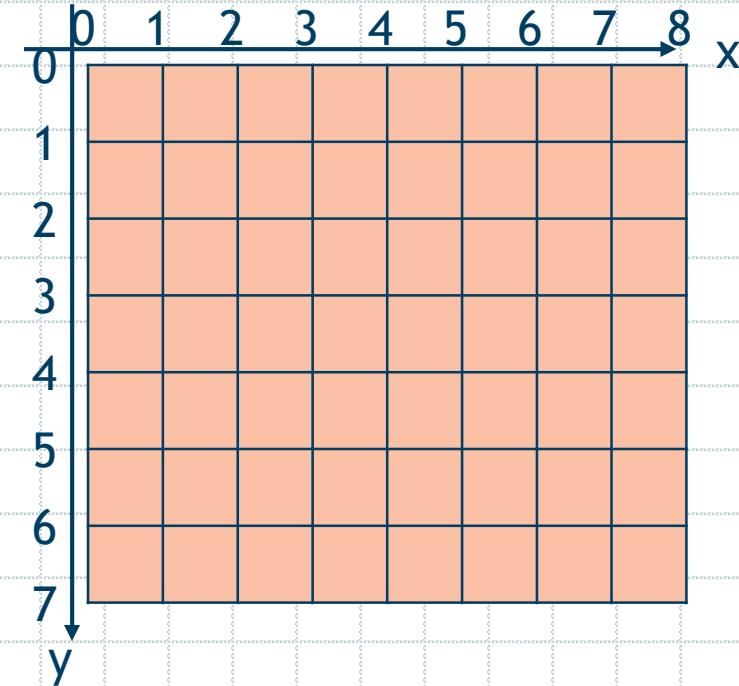
Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

- Pulau W1:
 - Ukuran 8x7
 - Terdapat 2 terminal

ILUSTRASI PERSOALAN





ITS

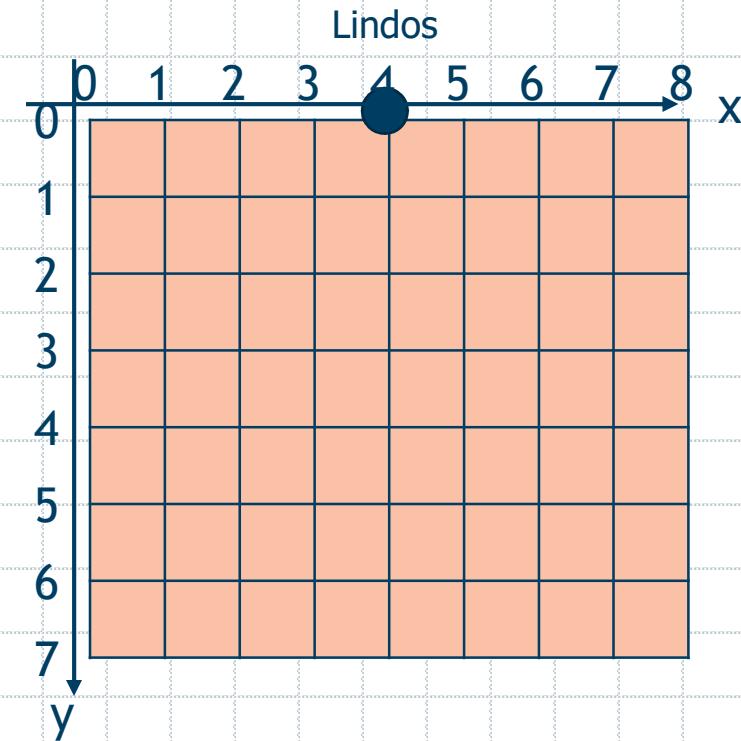
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W1:
 - Ukuran 8x7
 - Terdapat 2 terminal
 - ◆ Lindos 4 0

ILUSTRASI PERSOALAN





ITS

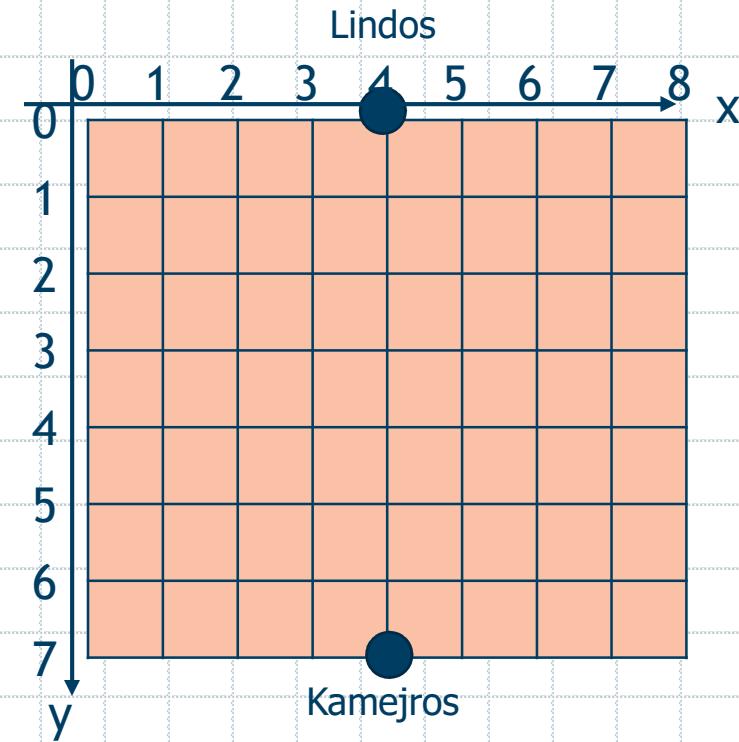
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W1:
 - Ukuran 8×7
 - Terdapat 2 terminal
 - ◆ Lindos 4 0
 - ◆ Kamejros 4 7

ILUSTRASI PERSOALAN





ITS

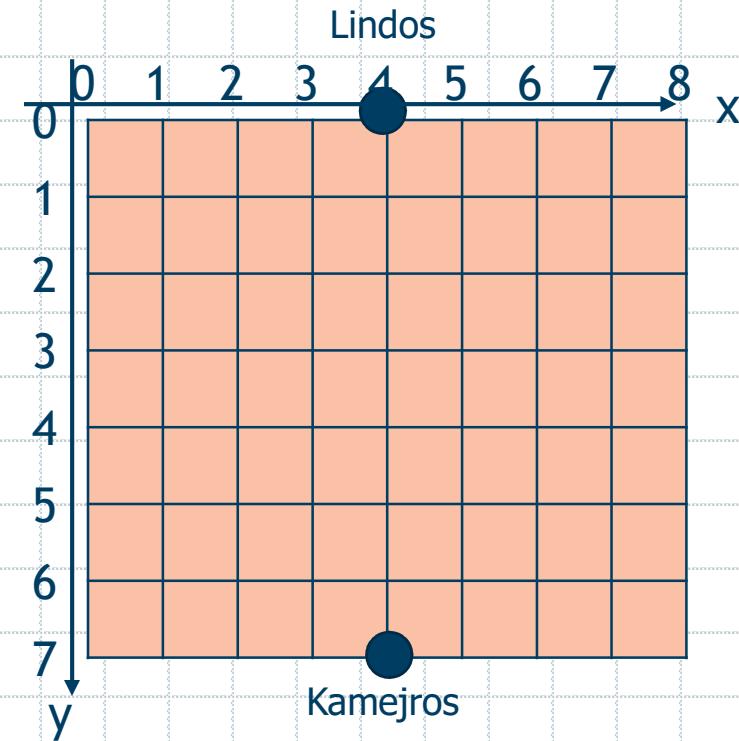
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W1:
 - Ukuran 8x7
 - Terdapat 2 terminal
 - ◆ Lindos 4 0
 - ◆ Kamejros 4 7
 - Terdapat 3 area terlarang

ILUSTRASI PERSOALAN





ITS

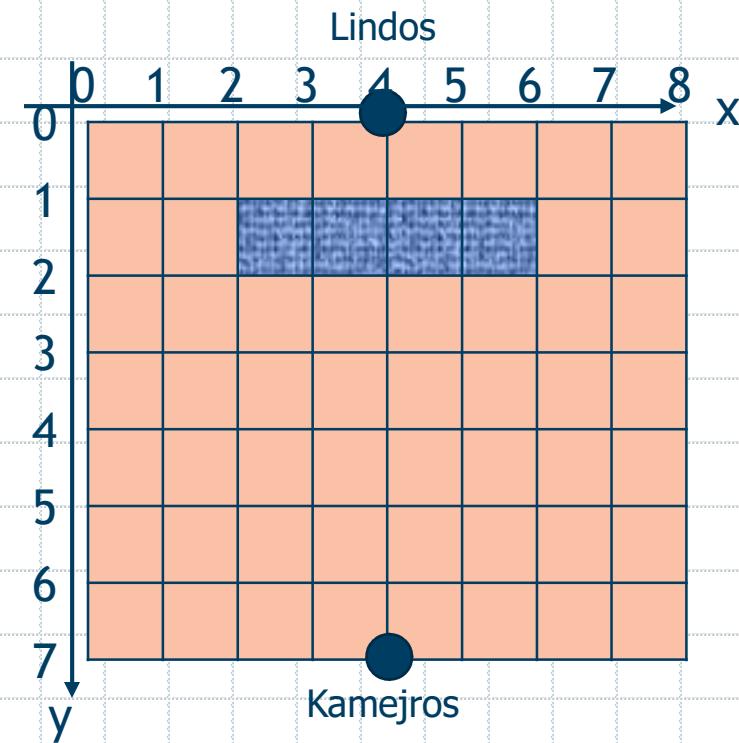
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W1:
 - Ukuran 8x7
 - Terdapat 2 terminal
 - ◆ Lindos 4 0
 - ◆ Kamejros 4 7
 - Terdapat 3 area terlarang
 - ◆ Area 1:
 - $(x_1, y_1): (2, 1)$
 - $(x_2, y_2): (6, 2)$

ILUSTRASI PERSOALAN





ITS

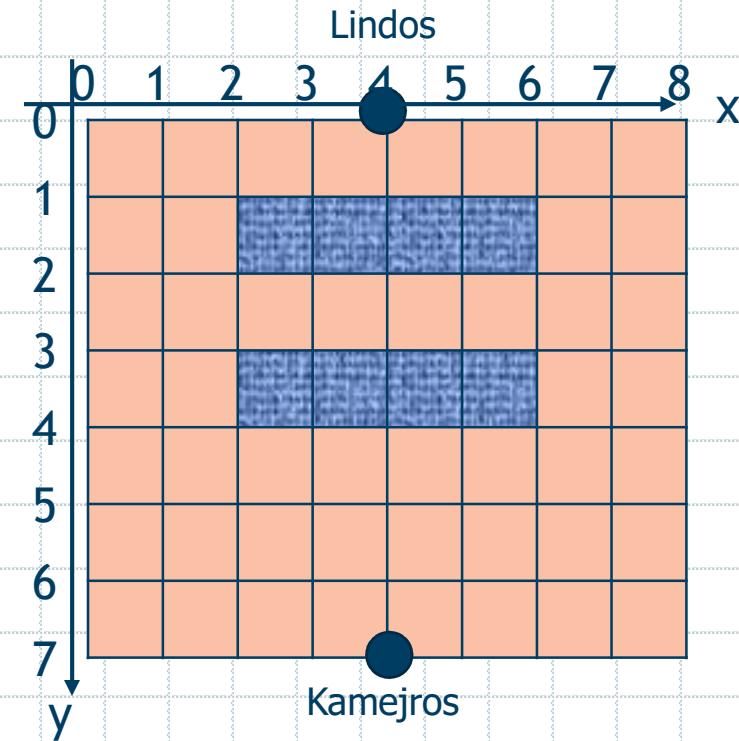
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W1:
 - Ukuran 8x7
 - Terdapat 2 terminal
 - ◆ Lindos 4 0
 - ◆ Kamejros 4 7
 - Terdapat 3 area terlarang
 - ◆ Area 1:
 - $(x_1, y_1): (2, 1)$
 - $(x_2, y_2): (6, 2)$
 - ◆ Area 2:
 - $(x_1, y_1): (2, 3)$
 - $(x_2, y_2): (6, 4)$

ILUSTRASI PERSOALAN





ITS

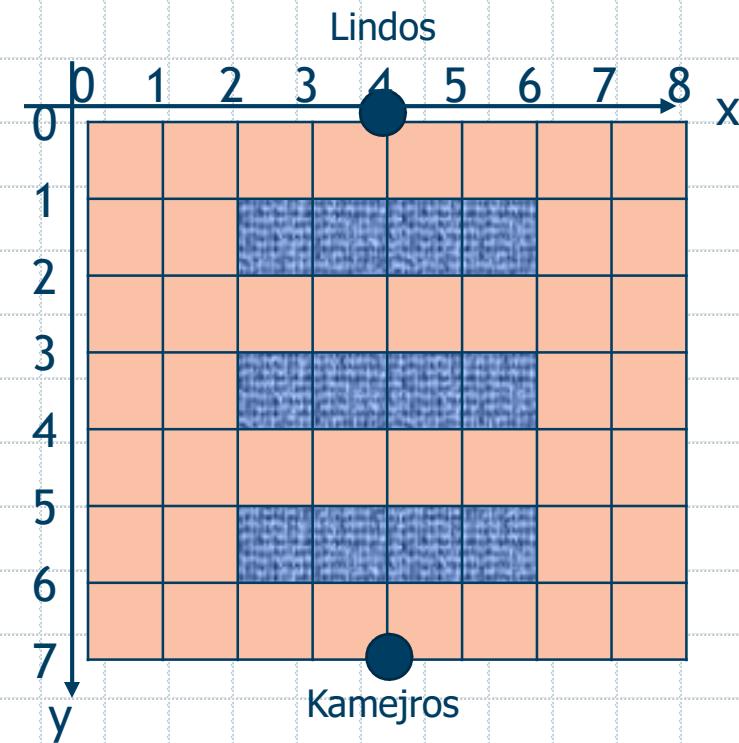
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W1:
 - Ukuran 8×7
 - Terdapat 2 terminal
 - ◆ Lindos 4 0
 - ◆ Kamejros 4 7
 - Terdapat 3 area terlarang
 - ◆ Area 1:
 - $(x_1, y_1): (2, 1)$
 - $(x_2, y_2): (6, 2)$
 - ◆ Area 2:
 - $(x_1, y_1): (2, 3)$
 - $(x_2, y_2): (6, 4)$
 - ◆ Area 3:
 - $(x_1, y_1): (2, 5)$
 - $(x_2, y_2): (6, 6)$

ILUSTRASI PERSOALAN





ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

Prinsip Dasar

Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

ILUSTRASI PERSOALAN

- Pulau W2:



ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

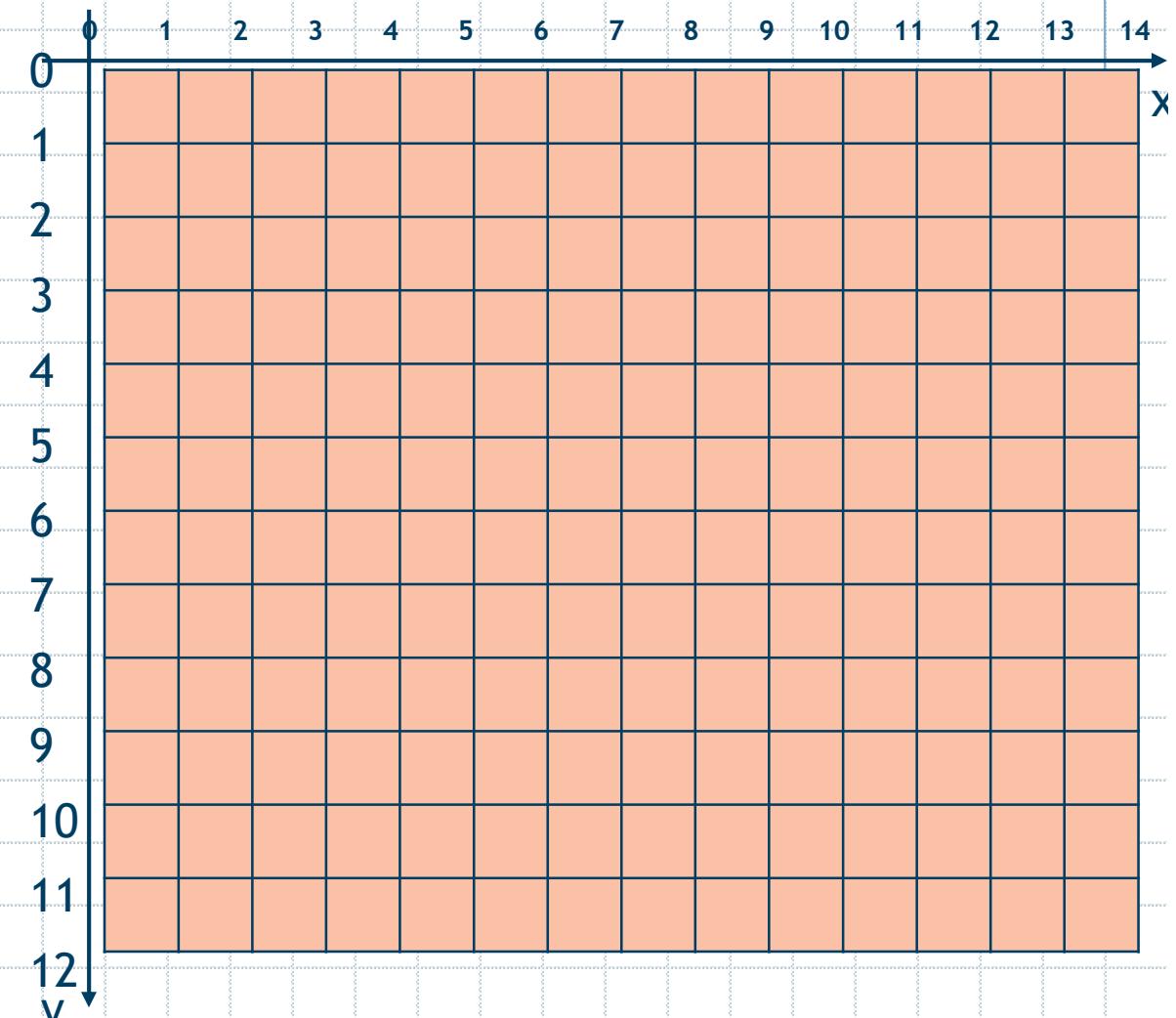
Prinsip Dasar

Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

- Pulau W2:
 - Ukuran 14x12





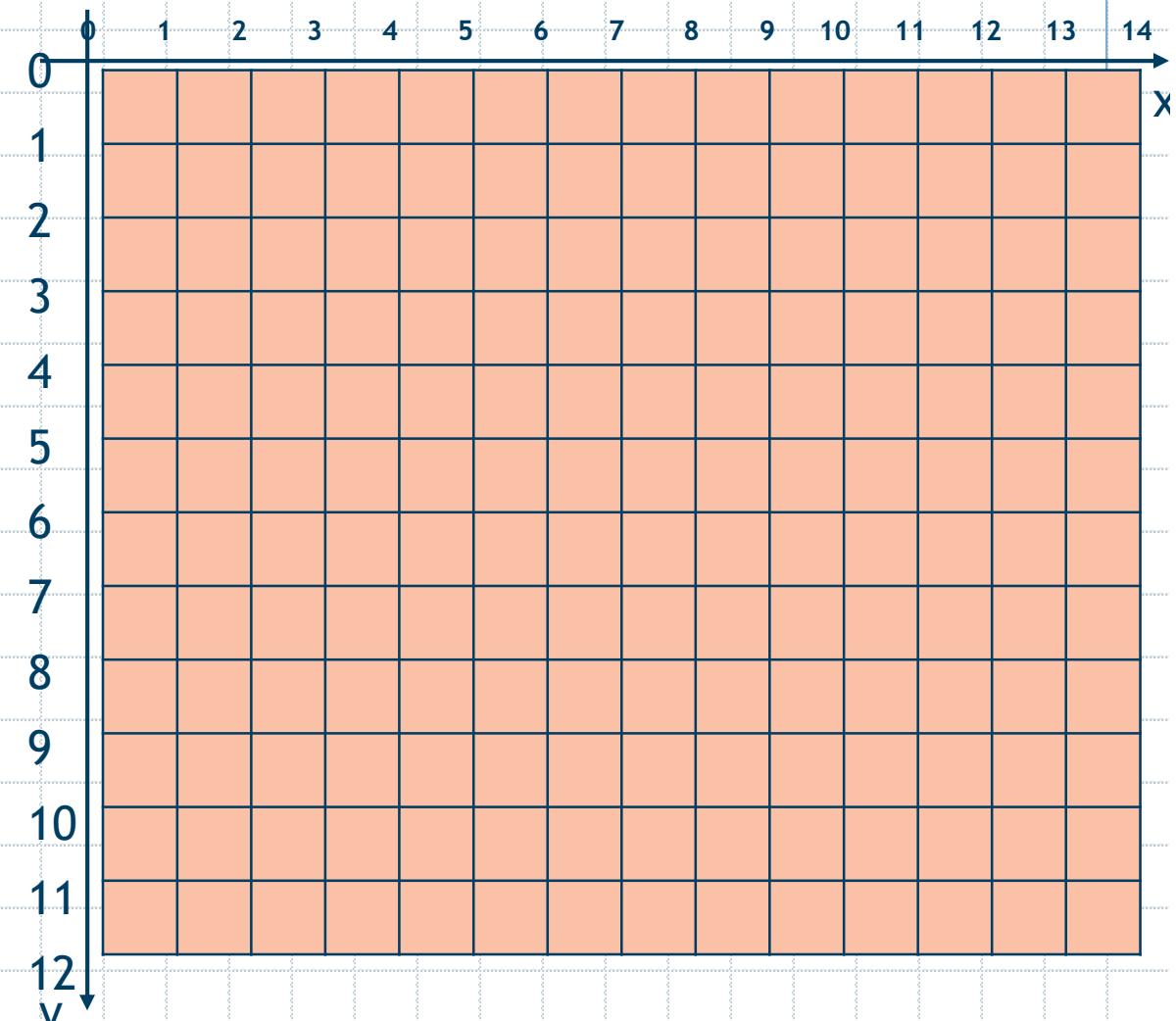
ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W2:
 - Ukuran 14x12
 - Terdapat 2 terminal:





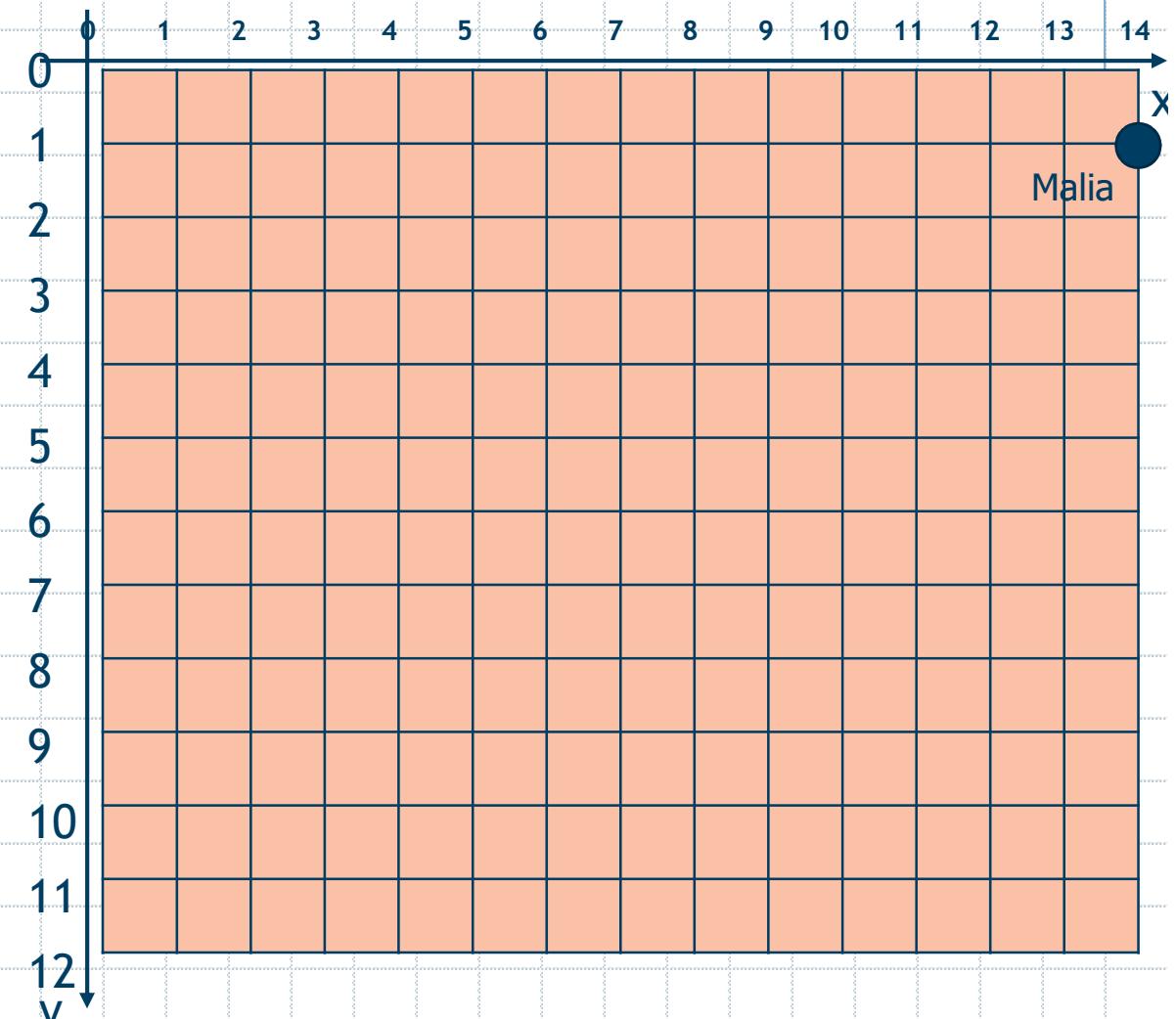
ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W2:
 - Ukuran 14x12
 - Terdapat 2 terminal:
 - ◆ Malia 14 1





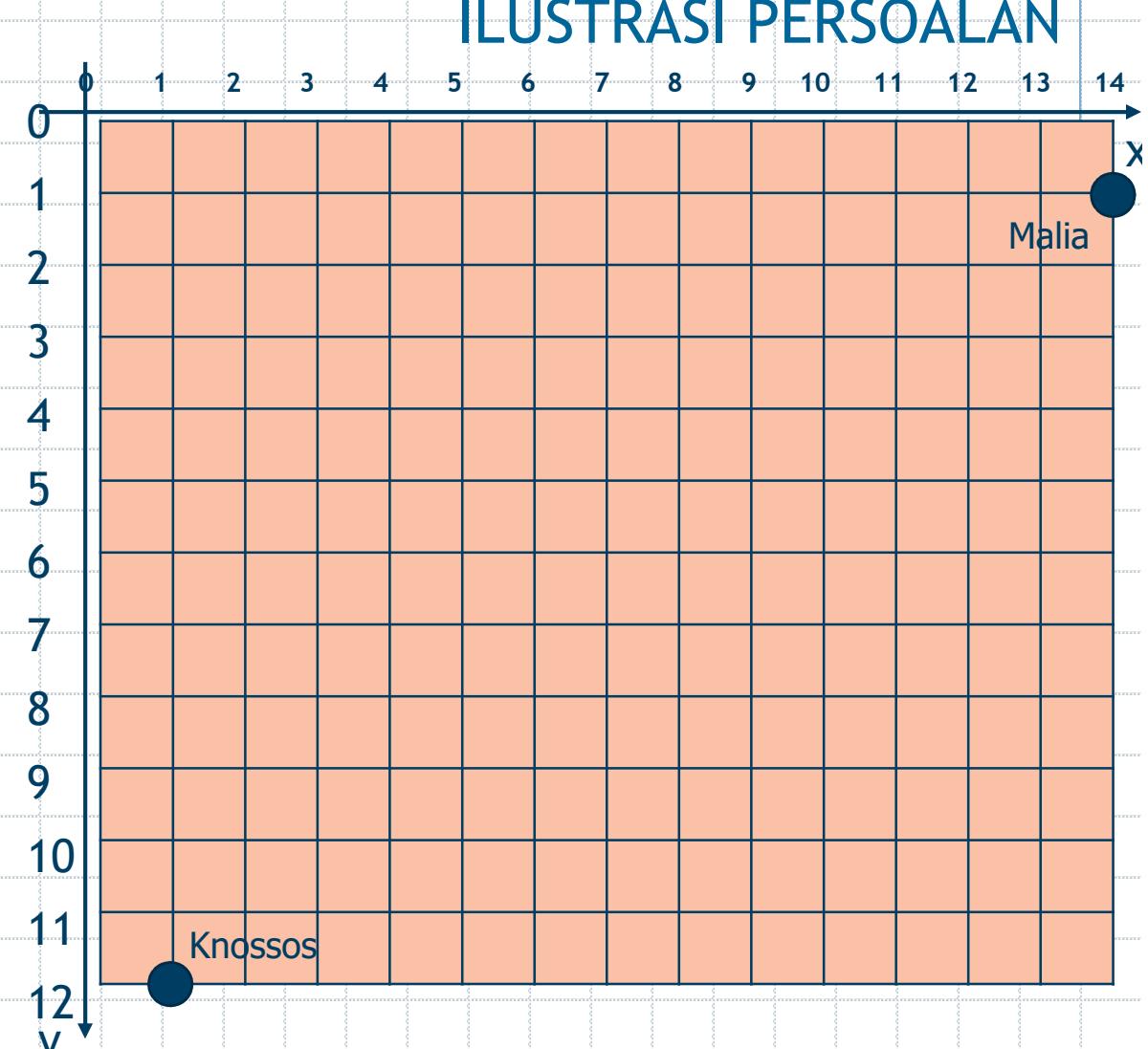
ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W2:
 - Ukuran 14x12
 - Terdapat 2 terminal:
 - ◆ Malia 14 1
 - ◆ Knossos 1 12





ITS

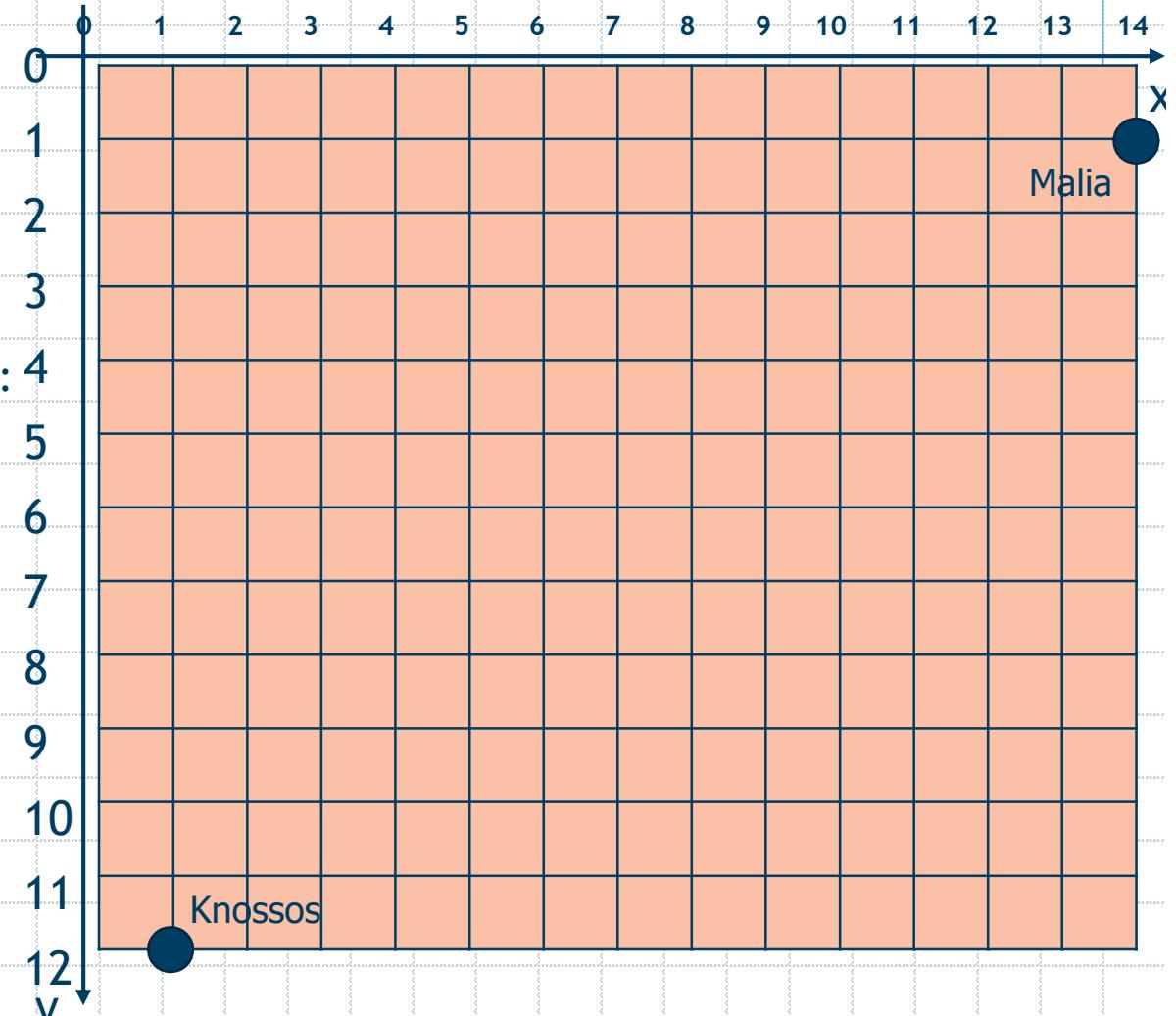
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- Pulau W2:
 - Ukuran 14x12
 - Terdapat 2 terminal:
 - ◆ Malia 14 1
 - ◆ Knossos 1 12
 - Terdapat 5 area terlarang:





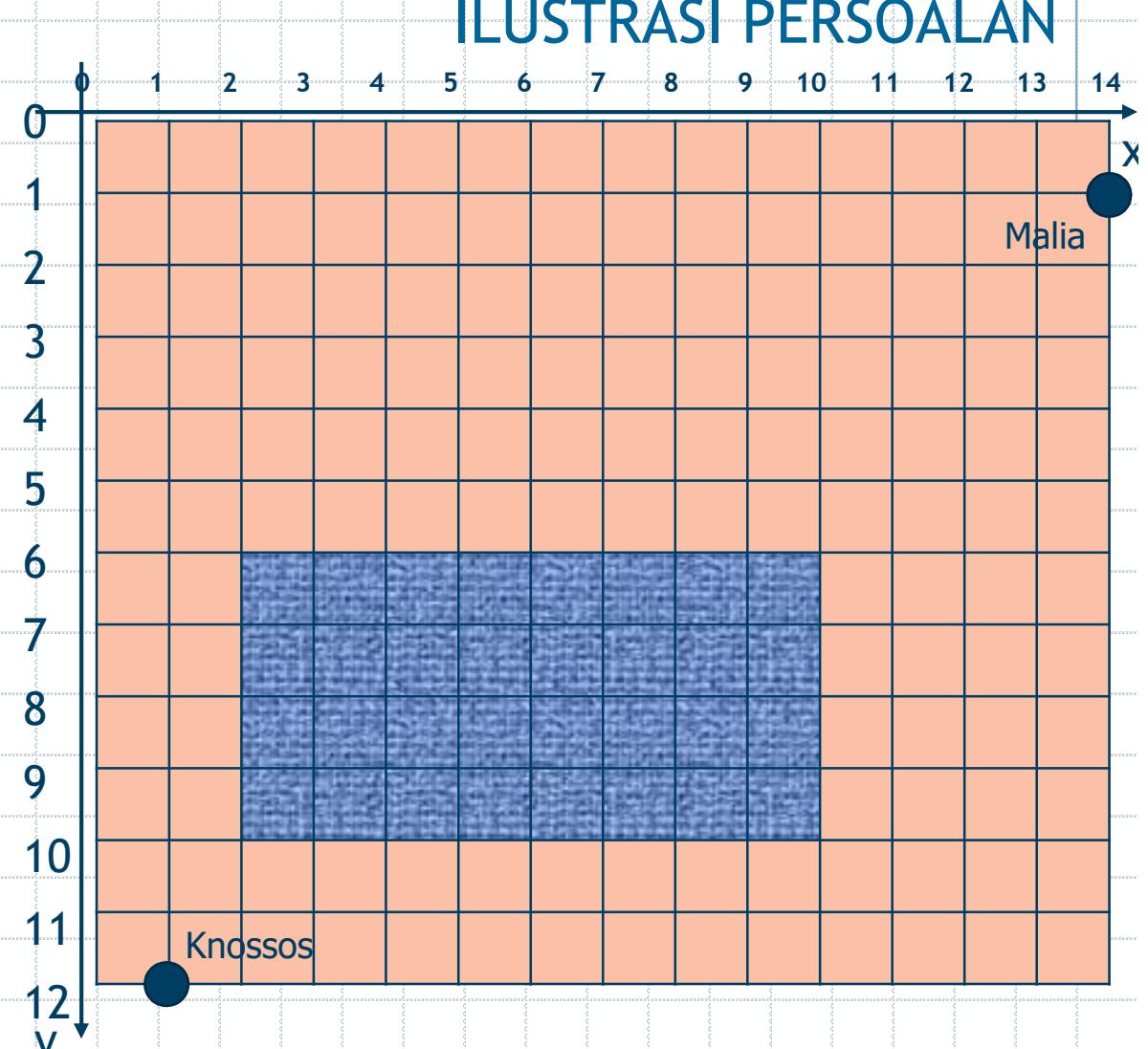
ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W2:
 - Ukuran 14×12
 - Terdapat 2 terminal:
 - ◆ Malia 14 1
 - ◆ Knossos 1 12
- Terdapat 5 area terlarang:
 - ◆ Area 1:
 - $(x_1, y_1): (2, 6)$
 - $(x_2, y_2): (10, 10)$





ITS

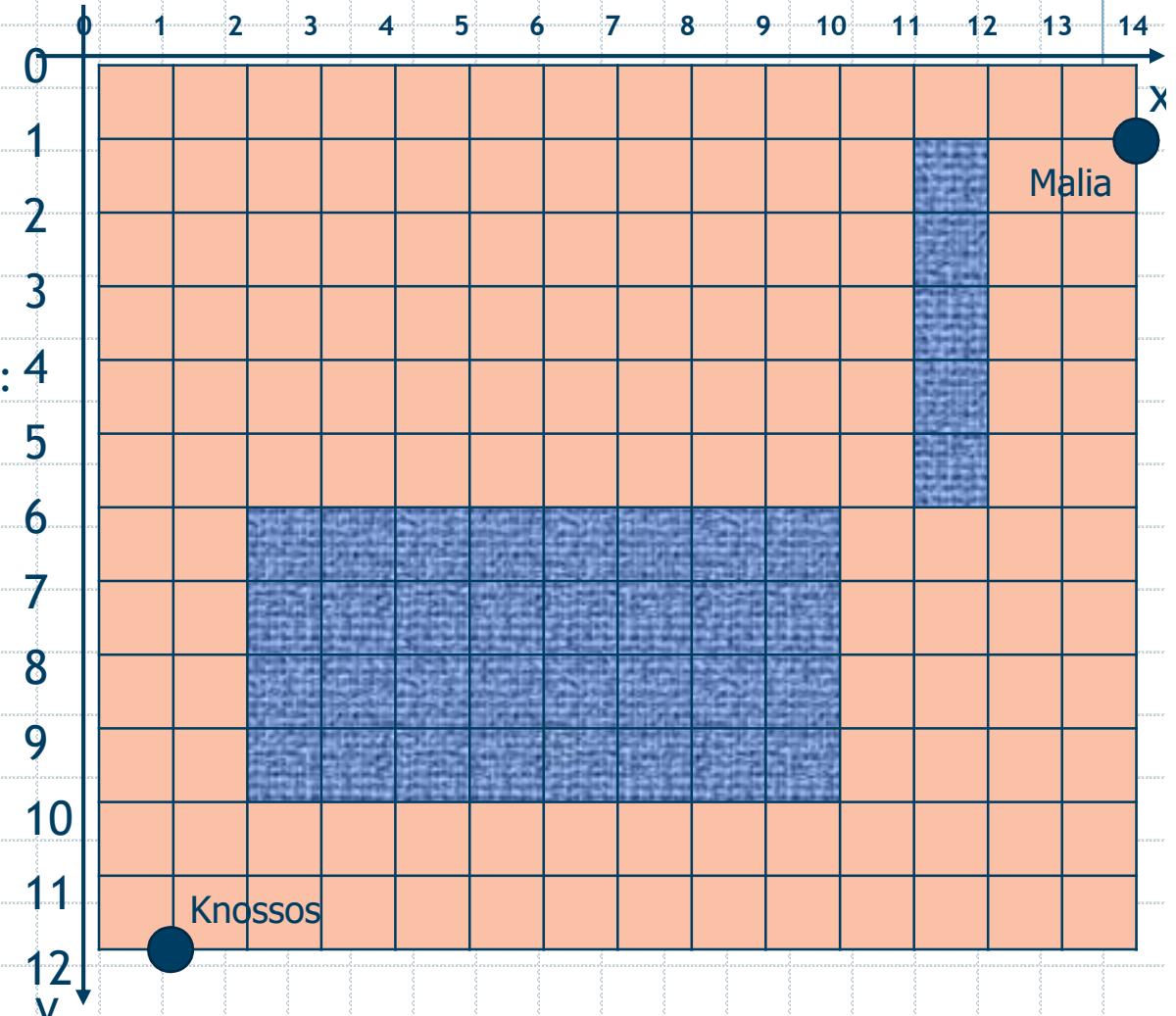
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- Pulau W2:
 - Ukuran 14×12
 - Terdapat 2 terminal:
 - ◆ Malia 14 1
 - ◆ Knossos 1 12
 - Terdapat 5 area terlarang:
 - ◆ Area 1:
 - $(x_1, y_1): (2, 6)$
 - $(x_2, y_2): (10, 10)$
 - ◆ Area 2:
 - $(x_1, y_1): (11, 1)$
 - $(x_2, y_2): (12, 6)$





ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

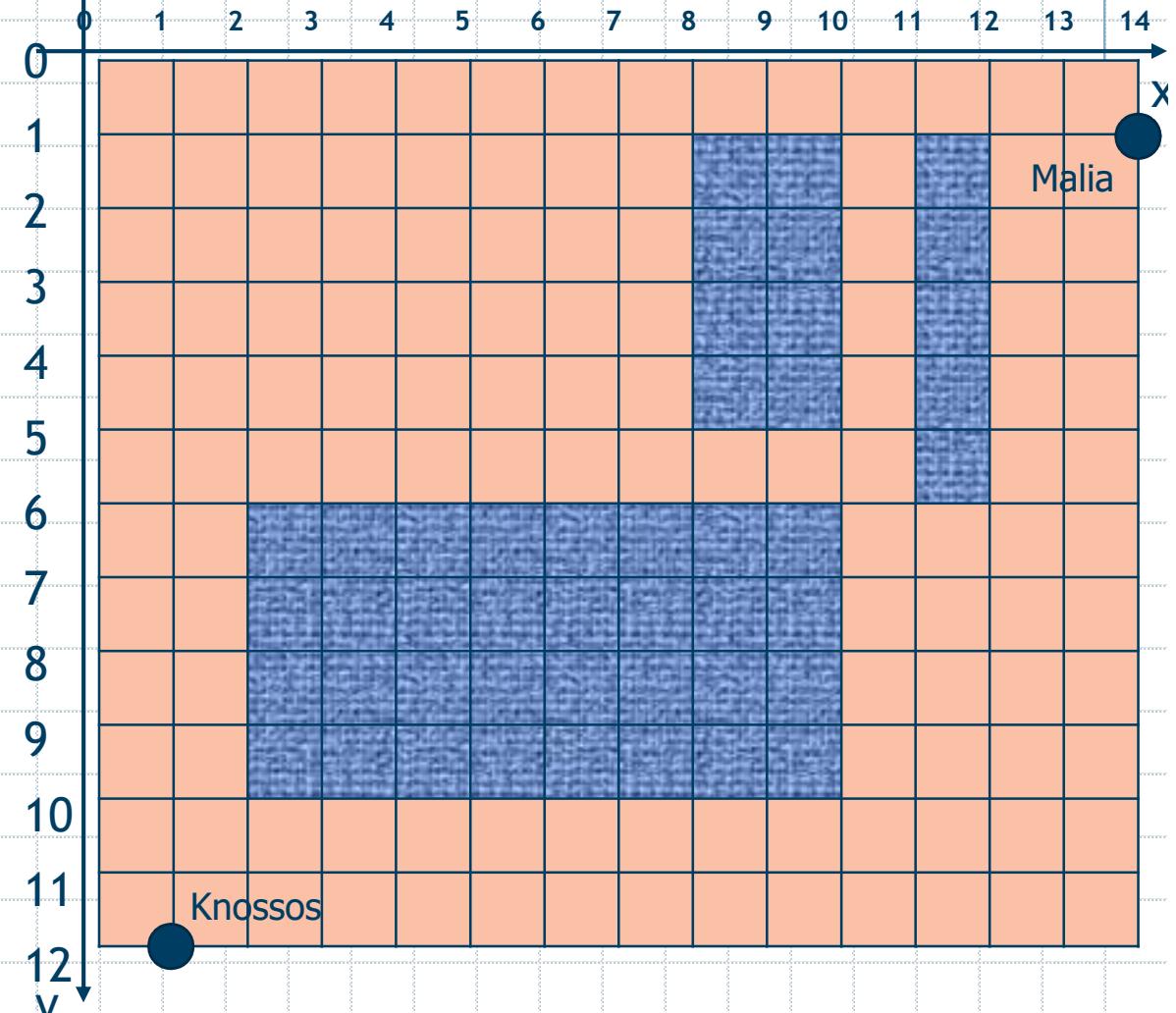
Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

Pulau W2:

- Ukuran 14x12
- Terdapat 2 terminal:
 - ◆ Malia 14 1
 - ◆ Knossos 1 12
- Terdapat 5 area terlarang:
 - ◆ Area 1:
 - $(x_1, y_1): (2, 6)$
 - $(x_2, y_2): (10, 10)$
 - ◆ Area 2:
 - $(x_1, y_1): (11, 1)$
 - $(x_2, y_2): (12, 6)$
 - ◆ Area 3:
 - $(x_1, y_1): (8, 1)$
 - $(x_2, y_2): (10, 5)$





ITS

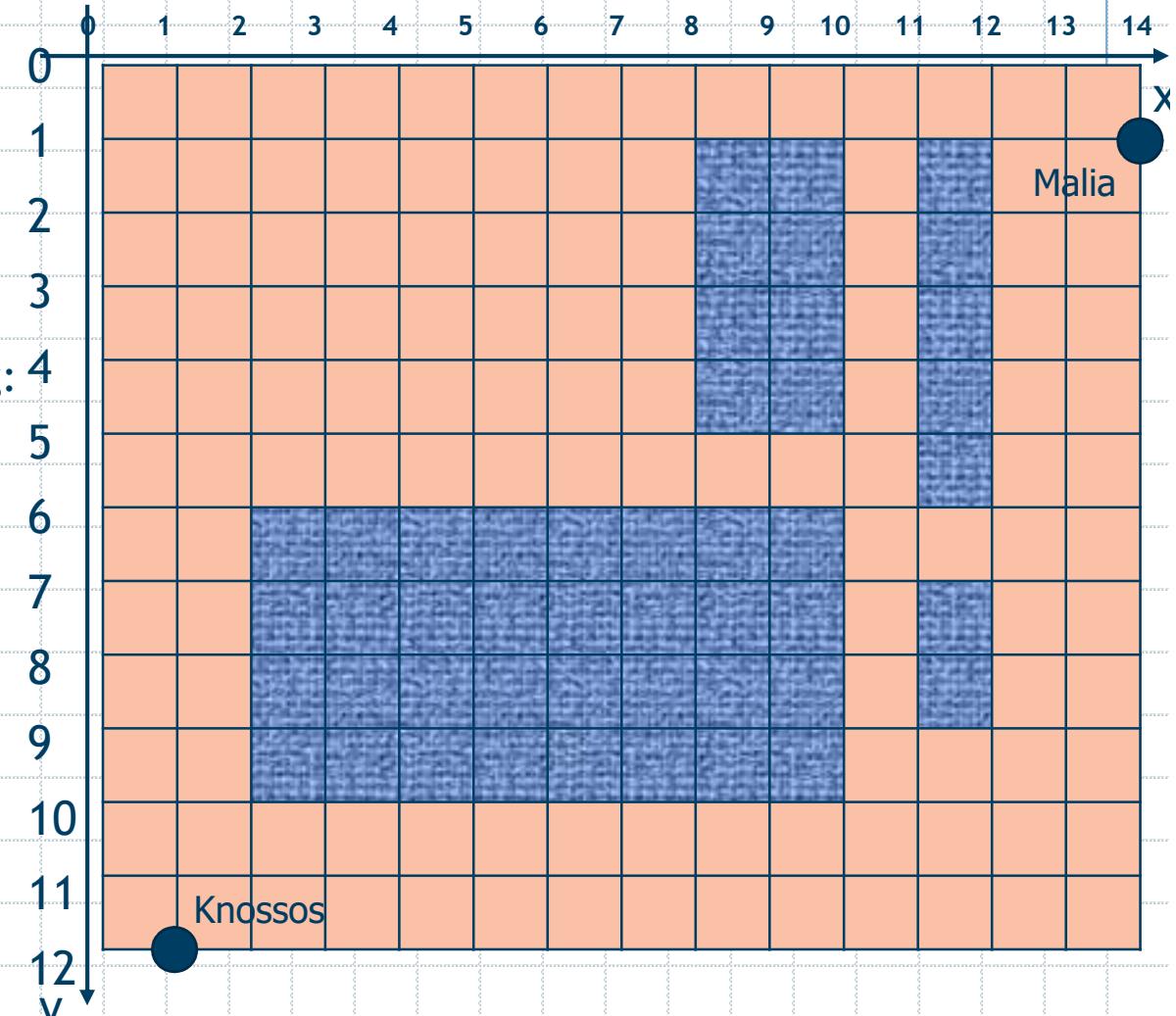
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- Pulau W2:
 - Ukuran 14x12
 - Terdapat 2 terminal:
 - ◆ Malia 14 1
 - ◆ Knossos 1 12
 - Terdapat 5 area terlarang:
 - ◆ Area 4:
 - $(x_1, y_1): (11, 7)$
 - $(x_2, y_2): (12, 9)$





ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

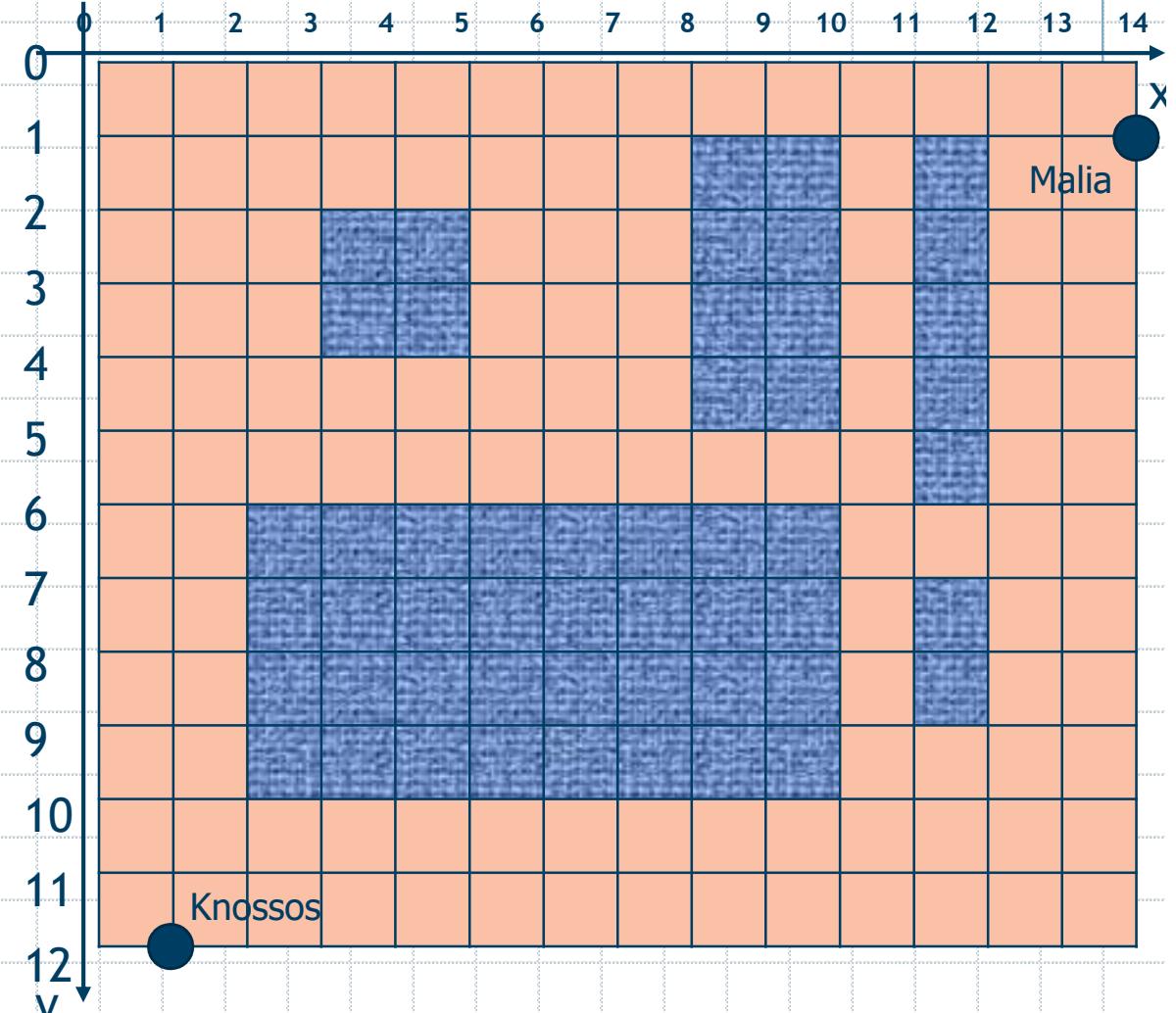
Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

Pulau W2:

- Ukuran 14x12
- Terdapat 2 terminal:
 - ◆ Malia 14 1
 - ◆ Knossos 1 12
- Terdapat 5 area terlarang:
 - ◆ Area 4:
 - $(x_1, y_1): (11, 7)$
 - $(x_2, y_2): (12, 9)$
 - ◆ Area 5:
 - $(x_1, y_1): (3, 2)$
 - $(x_2, y_2): (5, 4)$





ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

Prinsip Dasar

Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

- Pulau W3:

ILUSTRASI PERSOALAN



ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

Prinsip Dasar

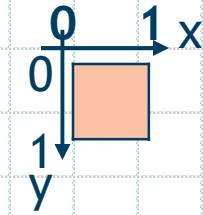
Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

- Pulau W3:
 - Ukuran 1×1
 - Terdapat 1 terminal:

ILUSTRASI PERSOALAN





ITS

Institut
Teknologi
Sepuluh Nopember

Visibility
Graph

Prinsip Dasar

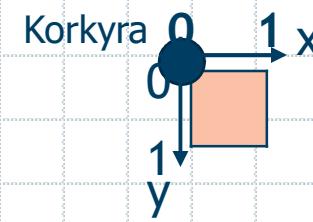
Visibility
Graph Naive

SPOJ The
Archipelago

Ilustrasi
Persoalan

- Pulau W3:
 - Ukuran 1x1
 - Terdapat 1 terminal:
 - ◆ Korkyra 0 0

ILUSTRASI PERSOALAN





ITS

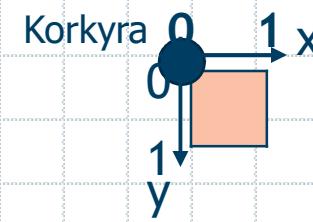
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

- Pulau W3:
 - Ukuran 1×1
 - Terdapat 1 terminal:
 - ◆ Korkyra 0 0
 - Terdapat 0 area terlarang

ILUSTRASI PERSOALAN





ITS

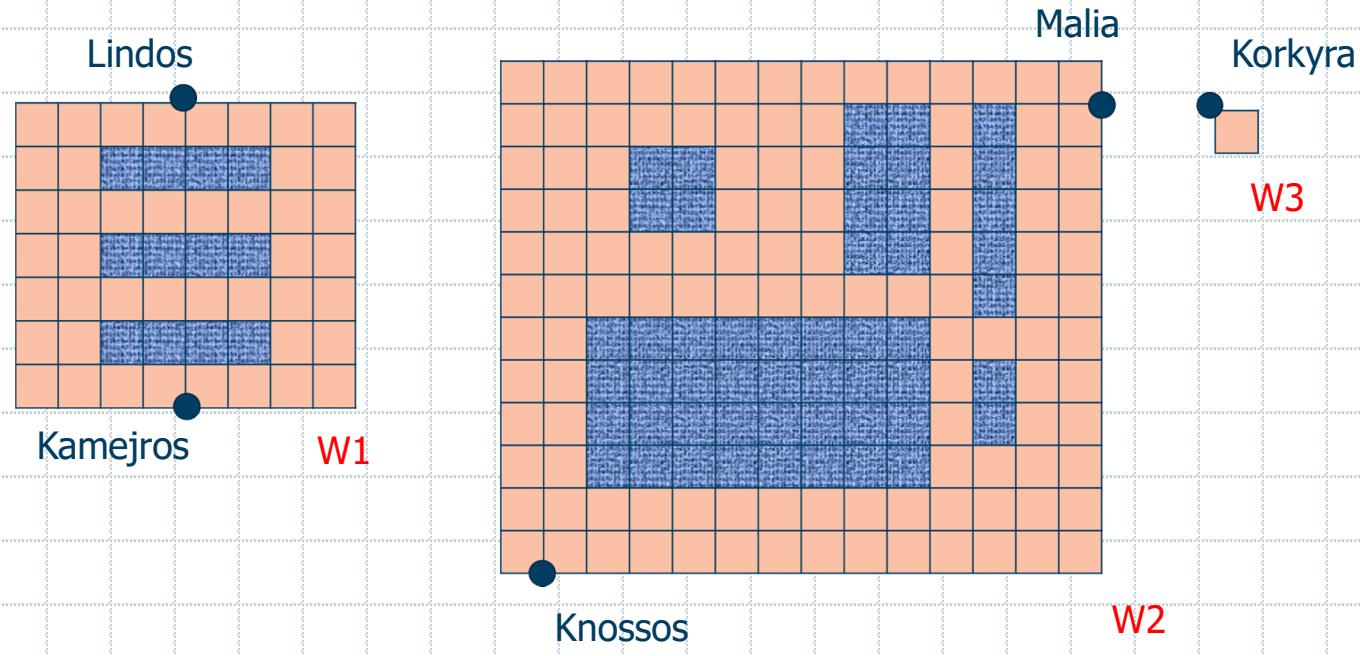
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- 2 jalur kapal ferry:





ITS

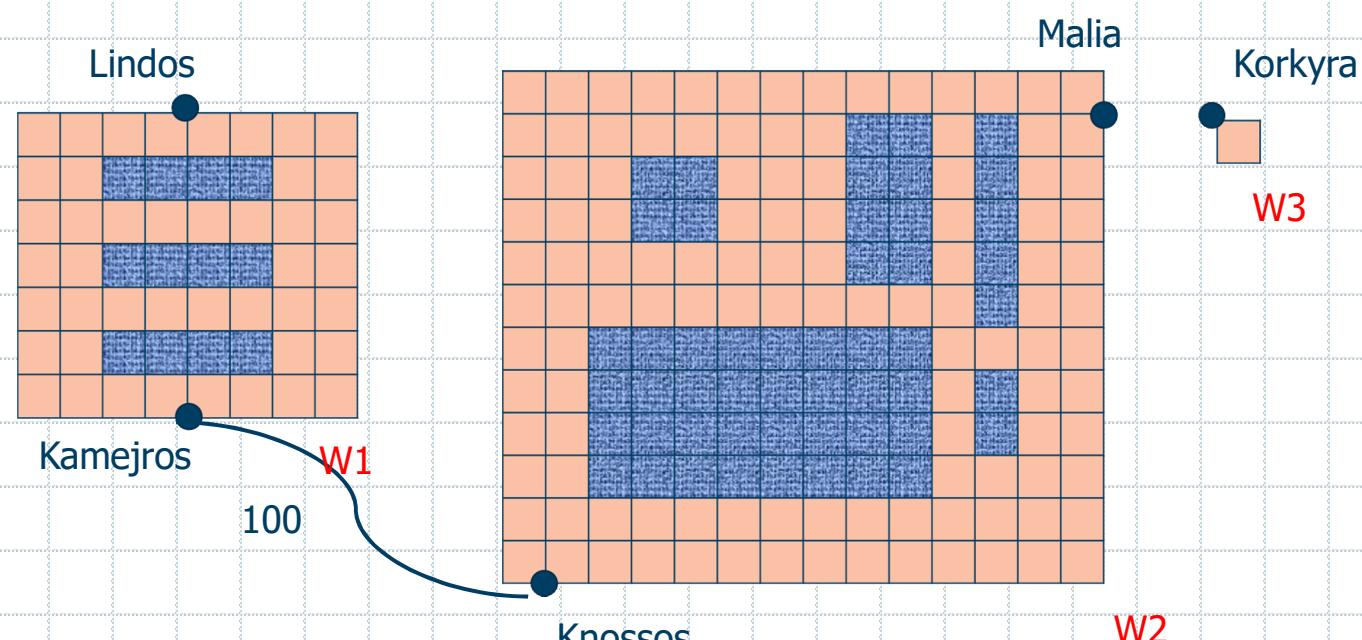
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- 2 jalur kapal ferry:
 - Kamejros W1 Knossos W2 100





ITS

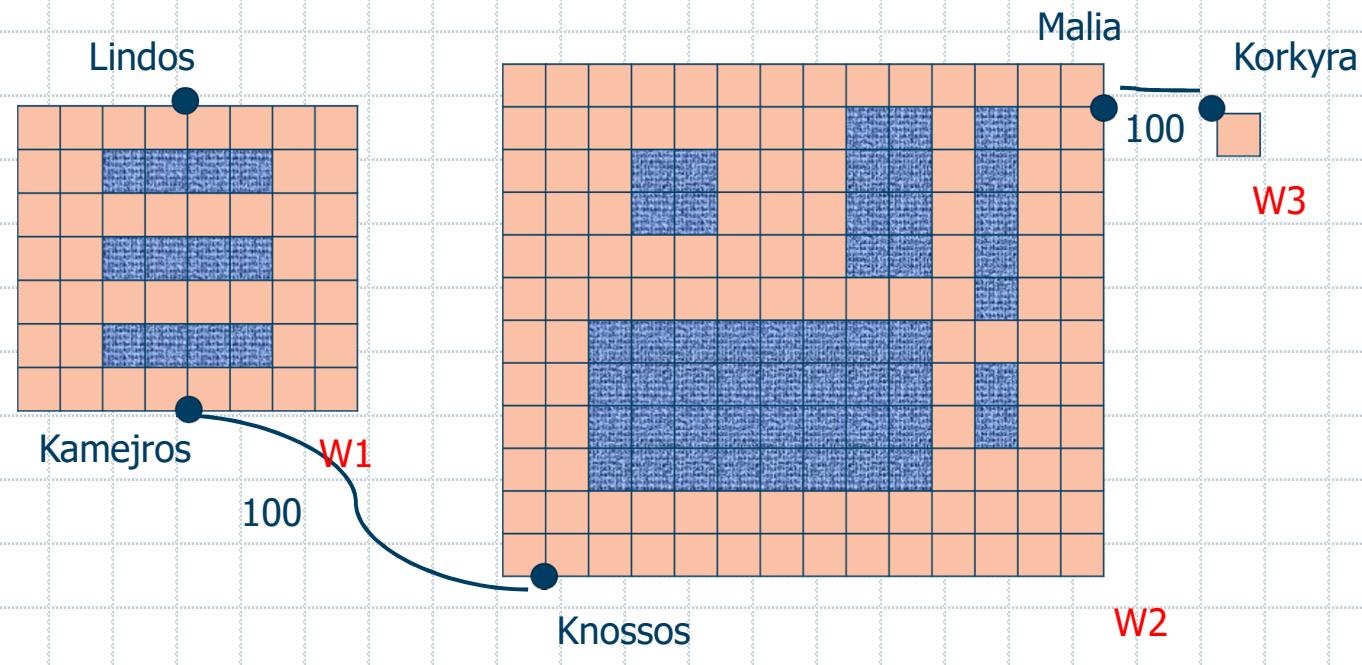
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- 2 jalur kapal ferry:
 - Kamejros W1 Knossos W2 100
 - Malia W2 Korkyra W3 100





ITS

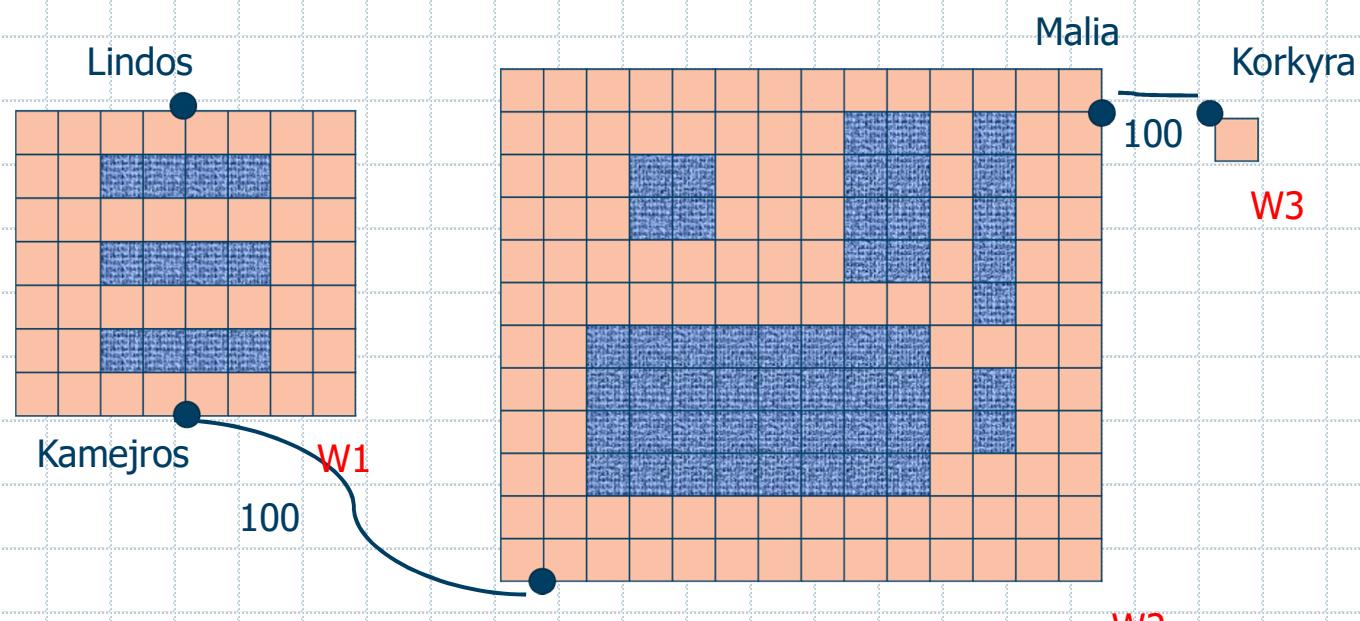
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- Permasalahan:
- Korkyra W3 ke Lindos W1 ?





ITS

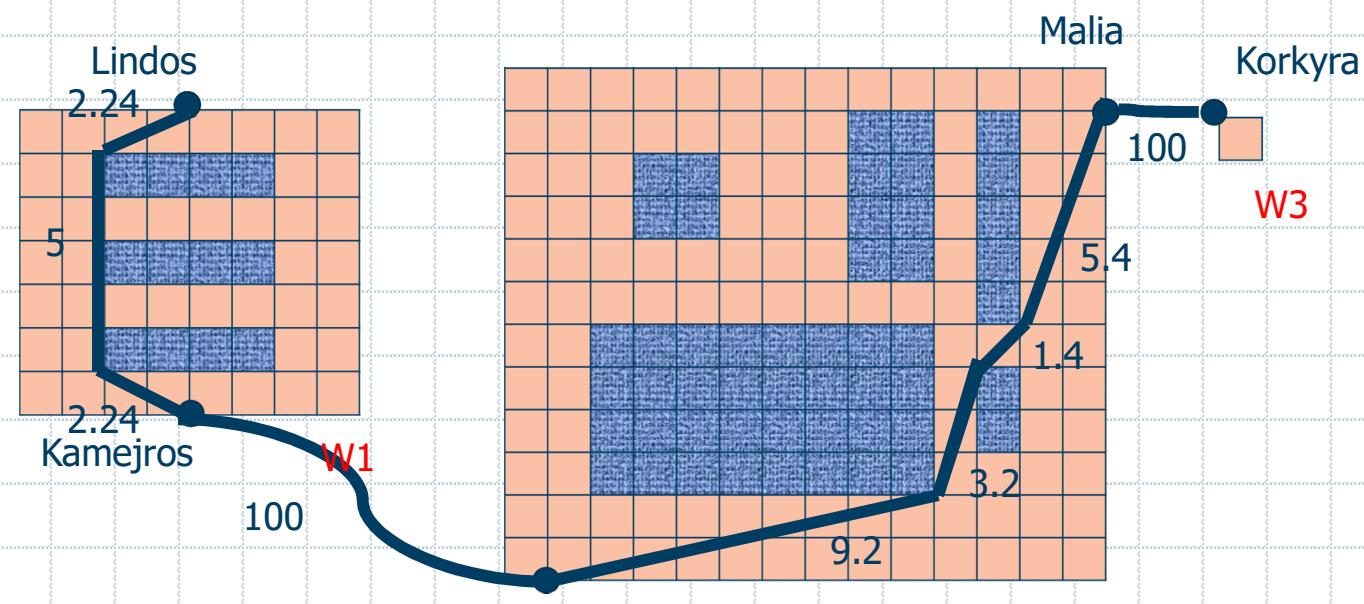
Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

- Hasil rute terpendeknya:





ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

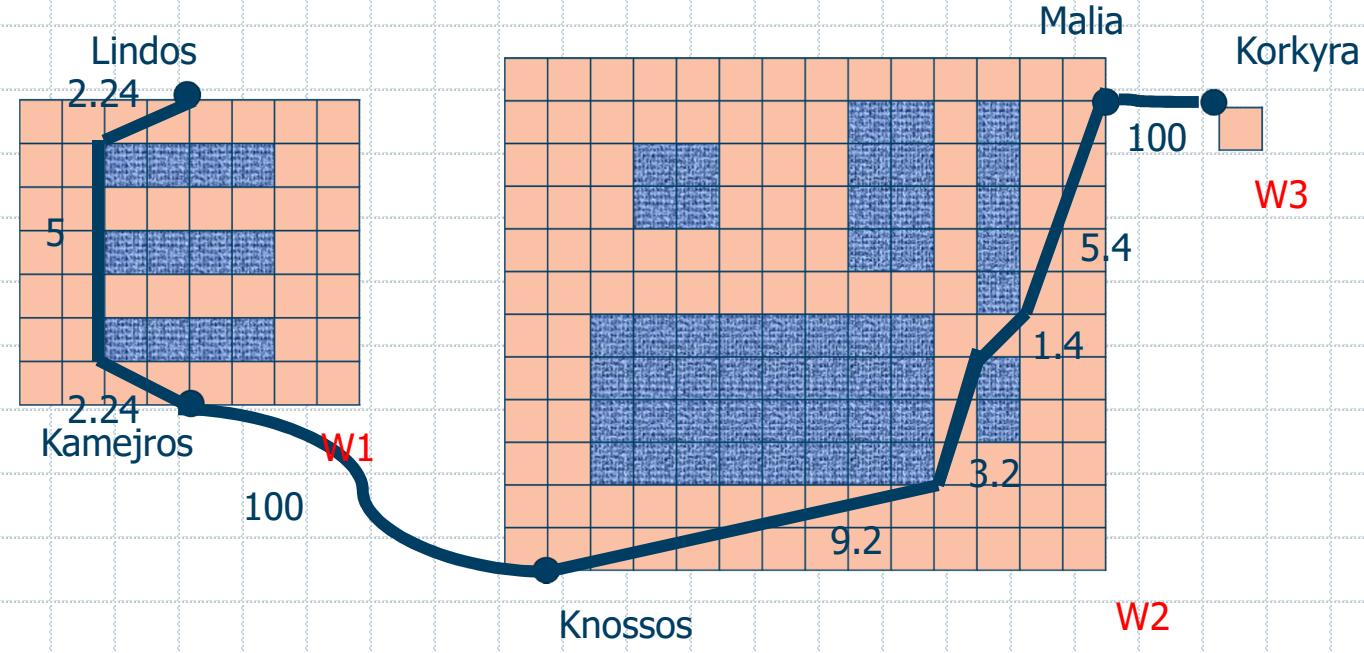
ILUSTRASI PERSOALAN

Total jarak pada pulau W1 : $2.24 + 5 + 2.24 = 9.48$

Total jarak pada pulau W2 : $5.4 + 1.4 + 3.2 + 9.2 = 19.2$

Total jarak pada pulau W3 : 0

Total jarak kapal ferry : $100 + 100 = 200$





ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

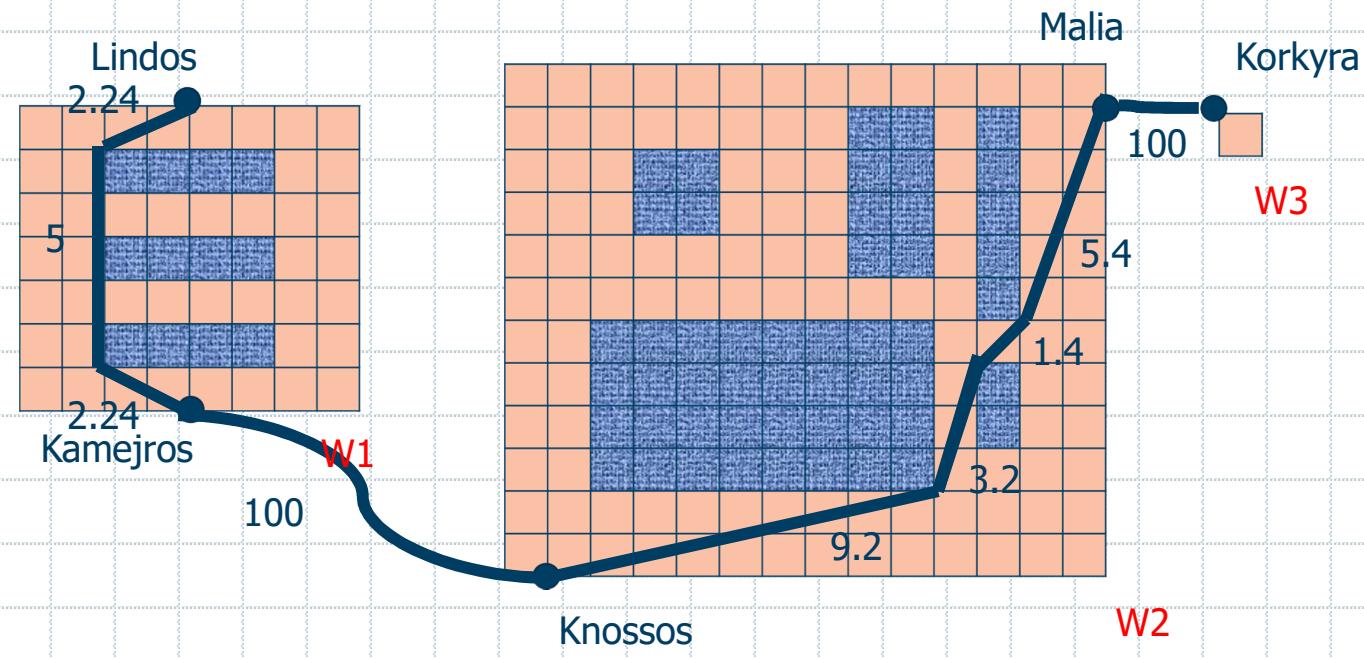
ILUSTRASI PERSOALAN

Total jarak pada pulau W1 : $2.24 + 5 + 2.24 = 9.48 \approx 10$

Total jarak pada pulau W2 : $5.4 + 1.4 + 3.2 + 9.2 = 19.2 \approx 20$

Total jarak pada pulau W3 : $0 \approx 0$

Total jarak kapal ferry : $100 + 100 = 200$





ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

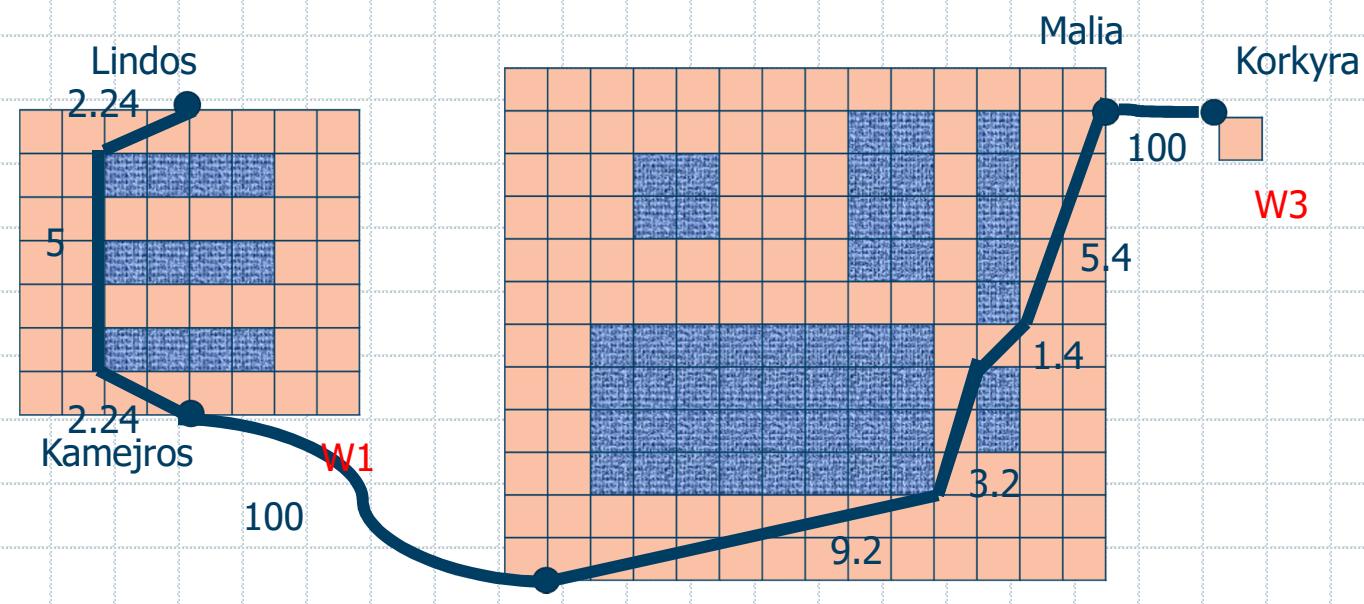
Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

Total jarak rute terpendek

$$: 10 + 20 + 0 + 200 = 230$$





ITS

Institut
Teknologi
Sepuluh NopemberVisibility
Graph

Prinsip Dasar

Visibility
Graph NaiveSPOJ The
ArchipelagoIlustrasi
Persoalan

ILUSTRASI PERSOALAN

Posisi yang dipilih:

Korkyra W3

Malia W2

12.6

11.7

10.10

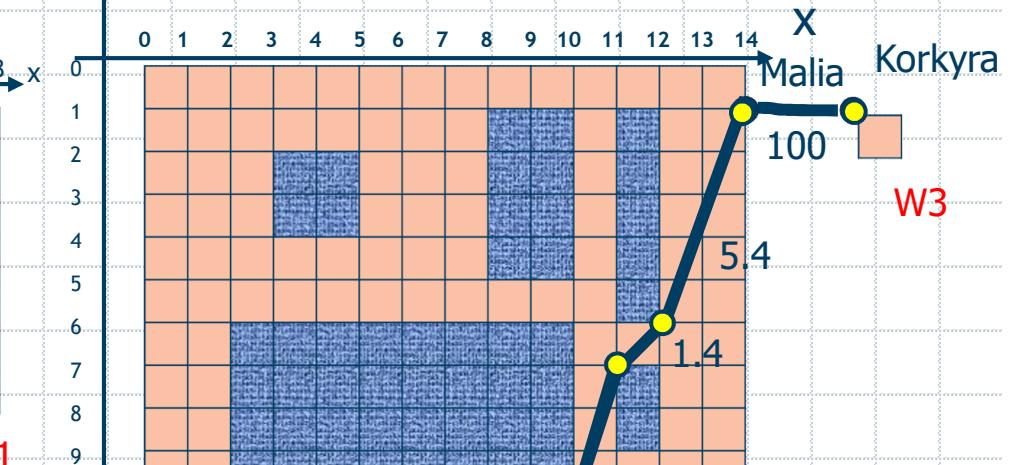
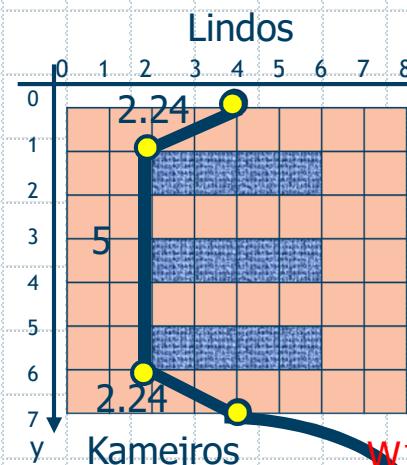
Knossos W2

Kamejros W1

2.6

2.1

Lindos W1





ITS

Institut
Teknologi
Sepuluh Nopember

PENDAHULUAN

TINJAUAN PUSTAKA

STRATEGI PENYELESAIAN

UJI COBA

KESIMPULAN



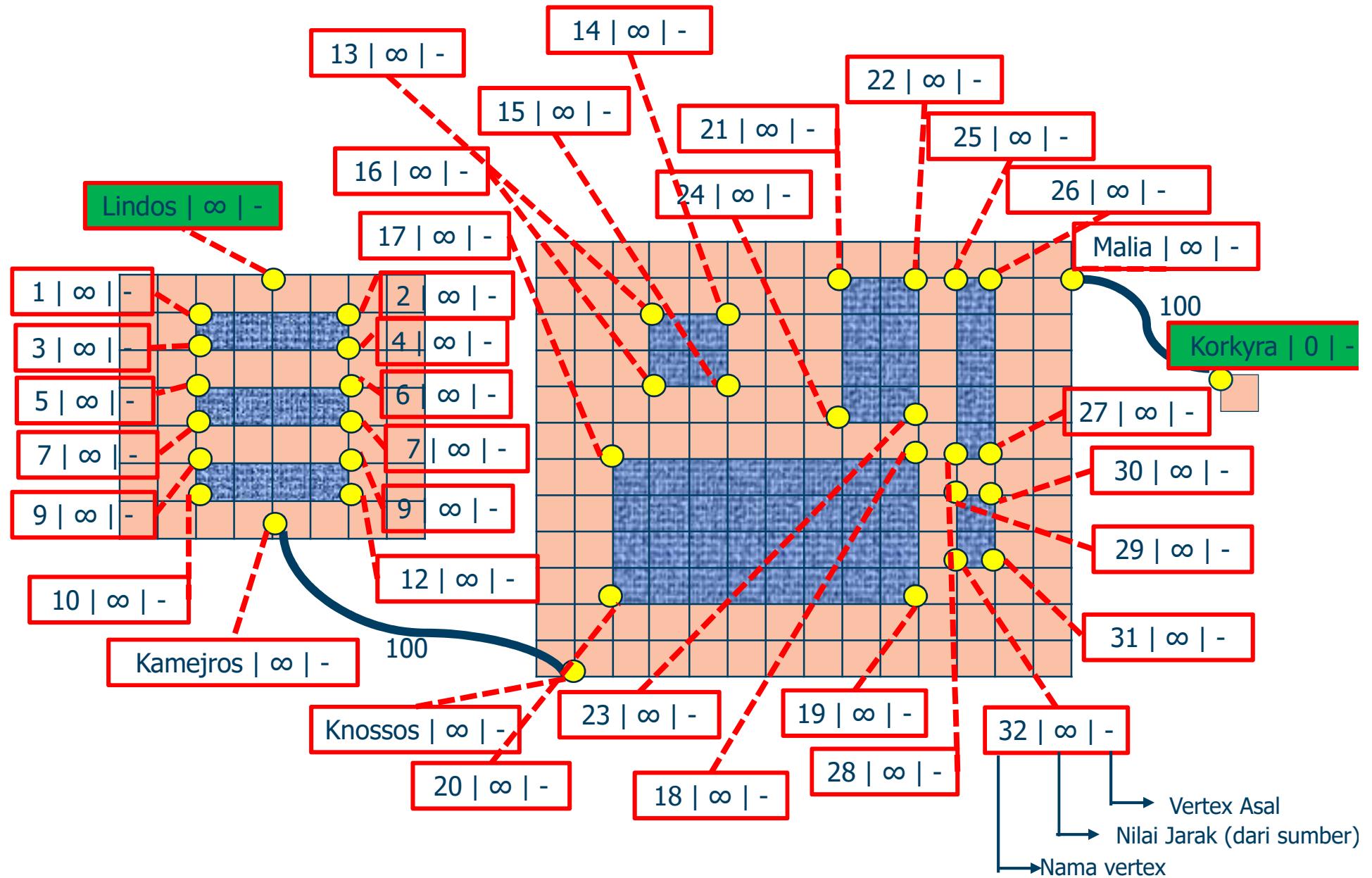
ITS

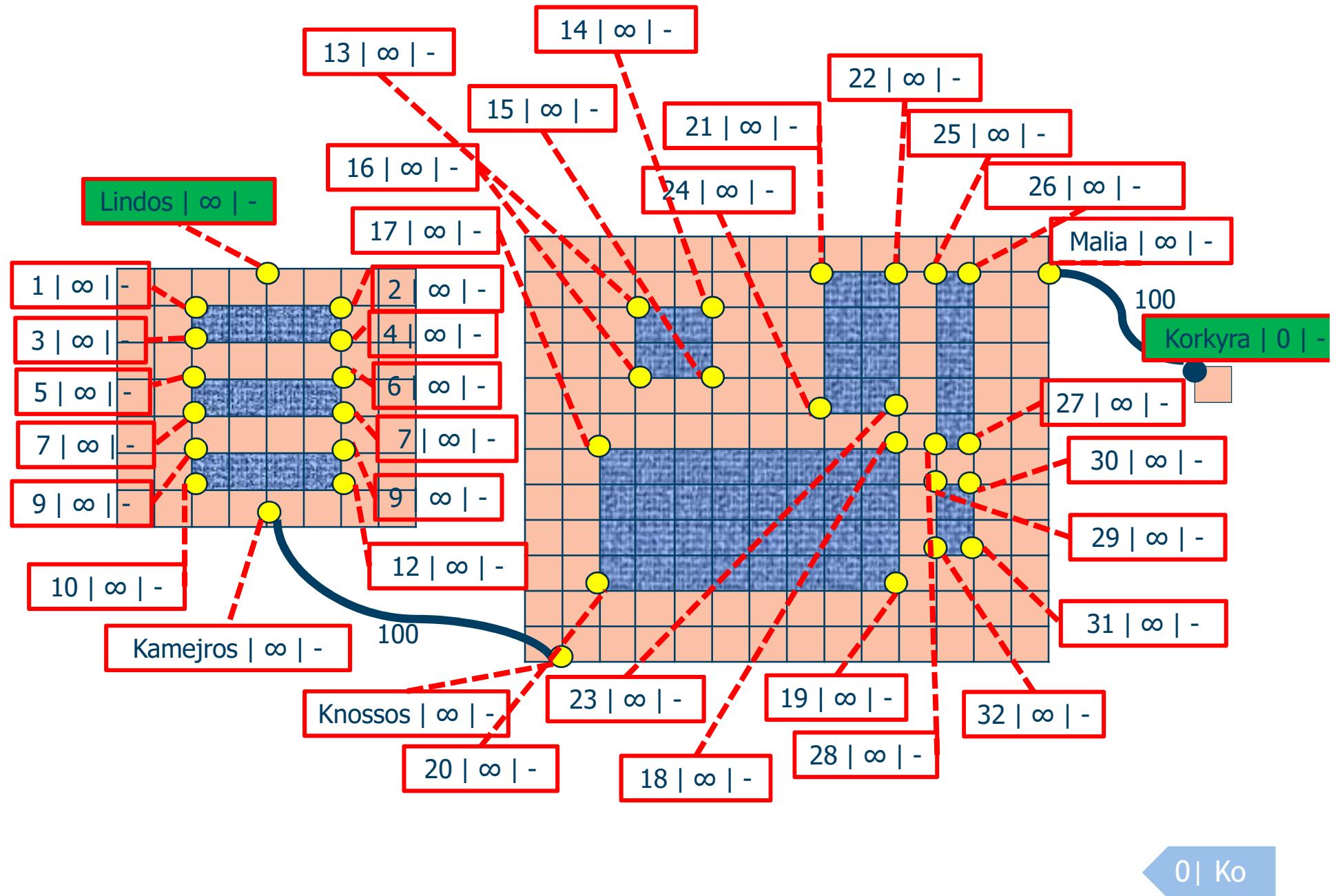
Institut
Teknologi
Sepuluh Nopember

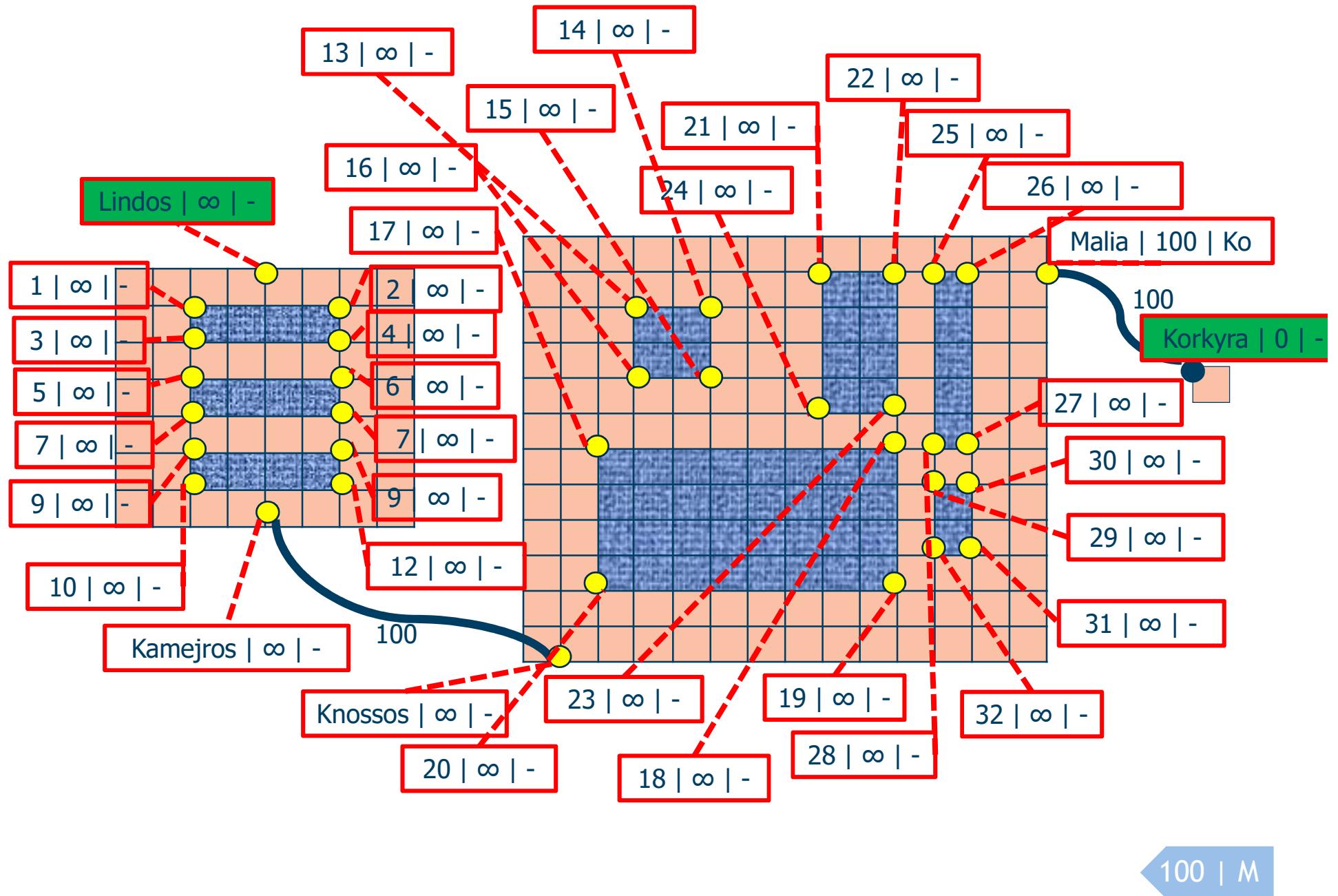
STRATEGI PENYELESAIAN

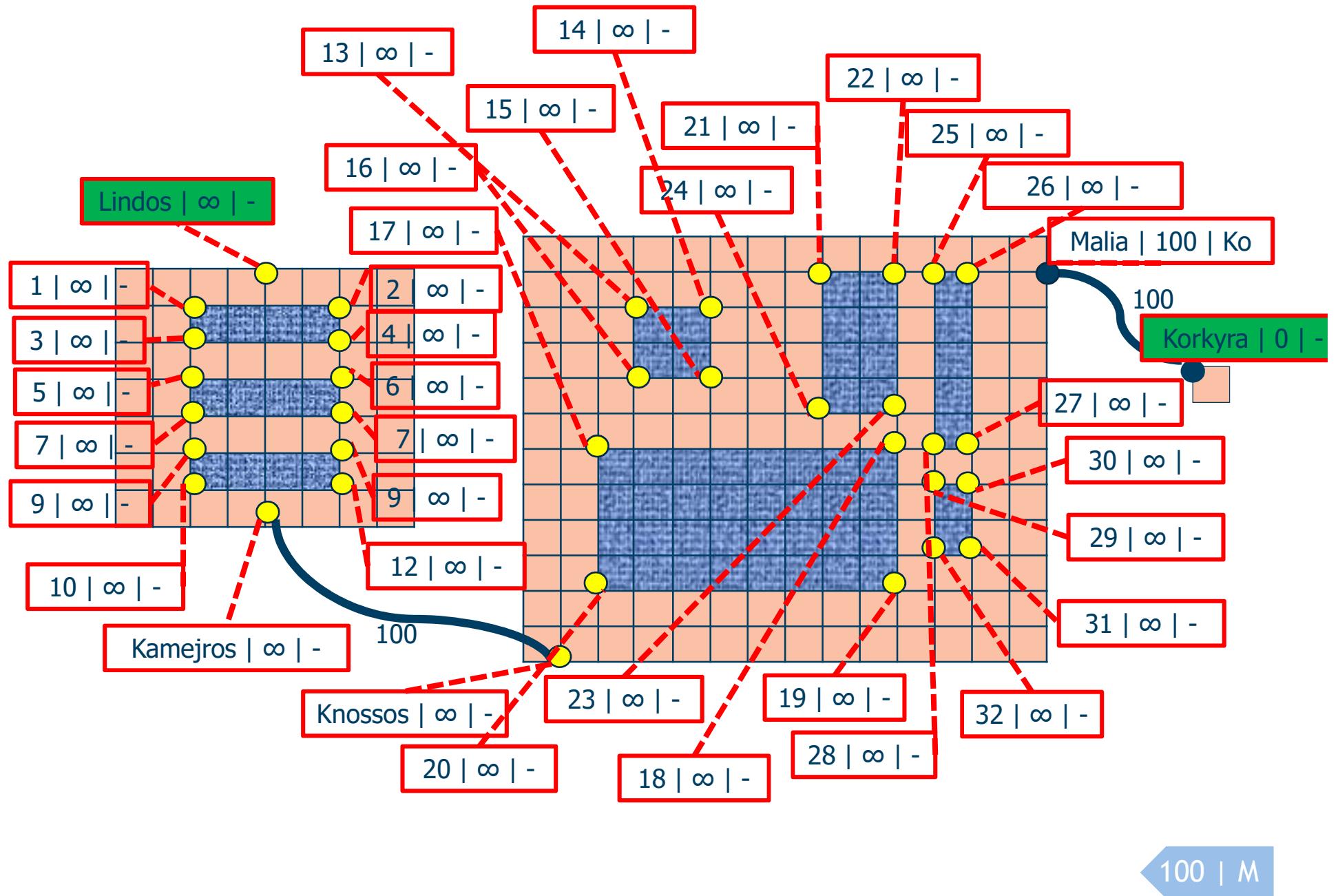
1. Penggunaan Visibility Graph Naive

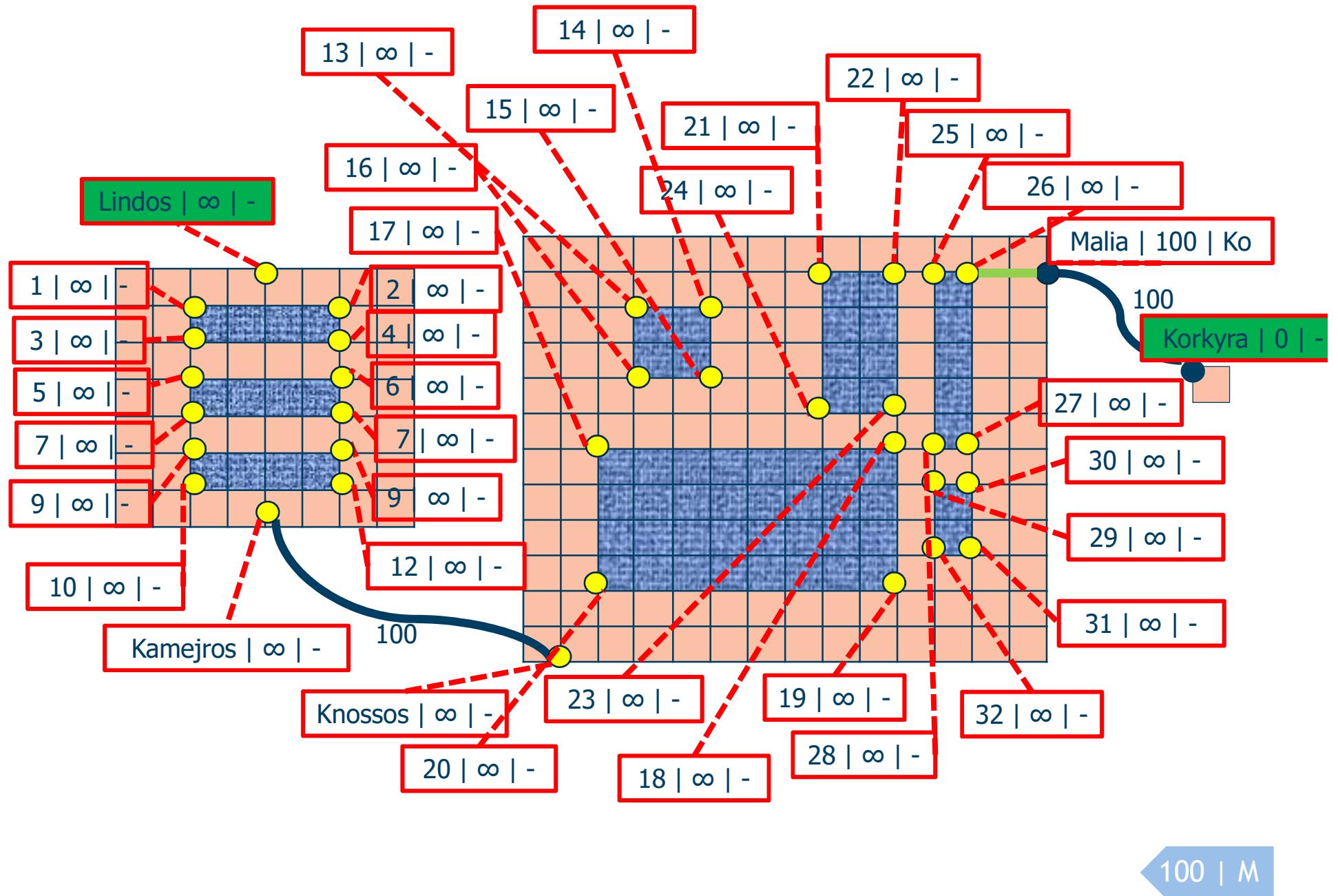
- Dikarenakan kompleksitas waktu yang cukup besar, yaitu $O(N^3)$, maka diperlukan efisiensi kinerja dari algoritma ini.
- Tidak semua *vertex* dicek visibilitasnya.
- Visibility graph naive dipanggil saat pengecekan *current_vertex* pada Dijkstra
- Tiap *vertex* hanya akan dicek visibilitasnya terhadap *vertex* lain se-pulau.

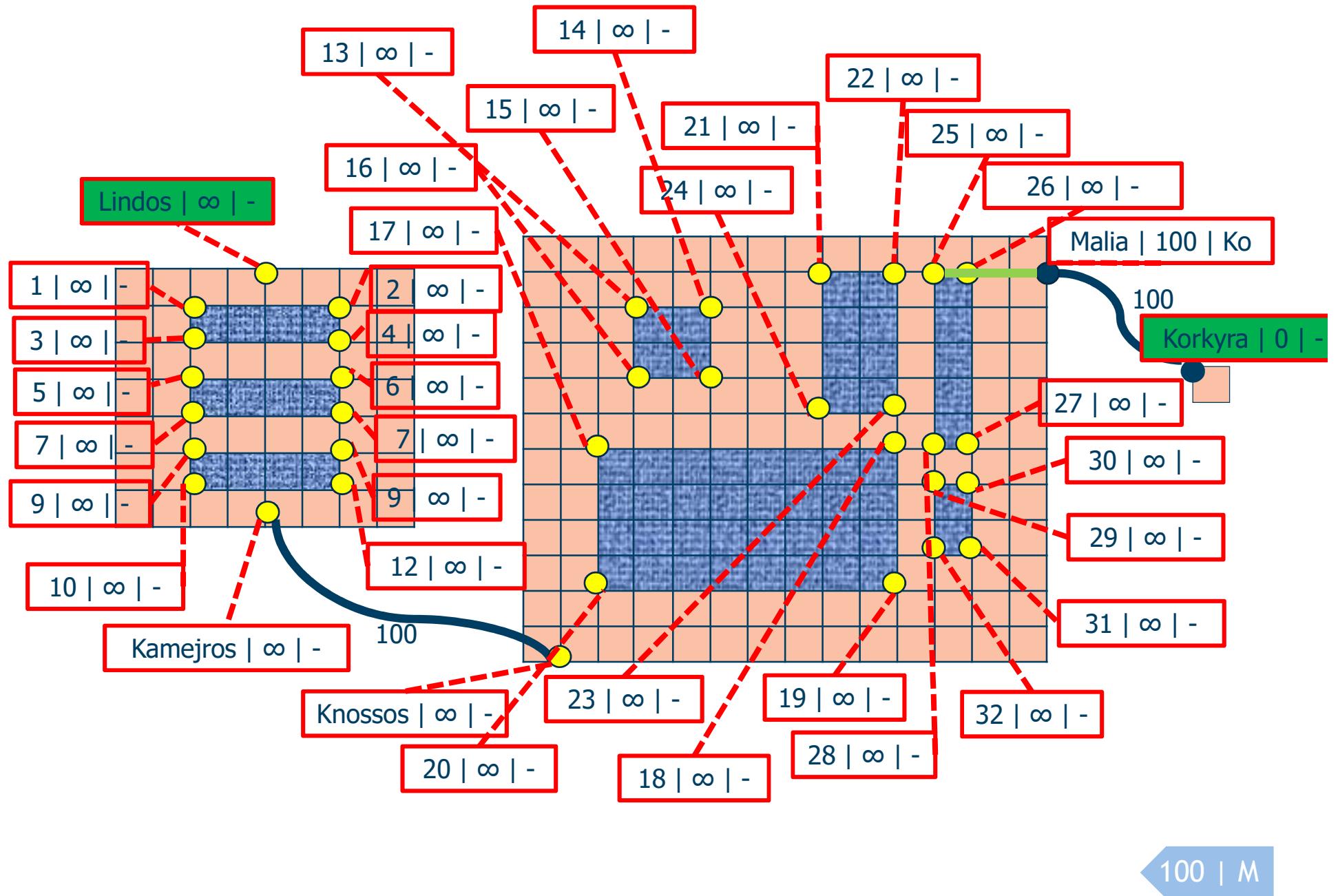


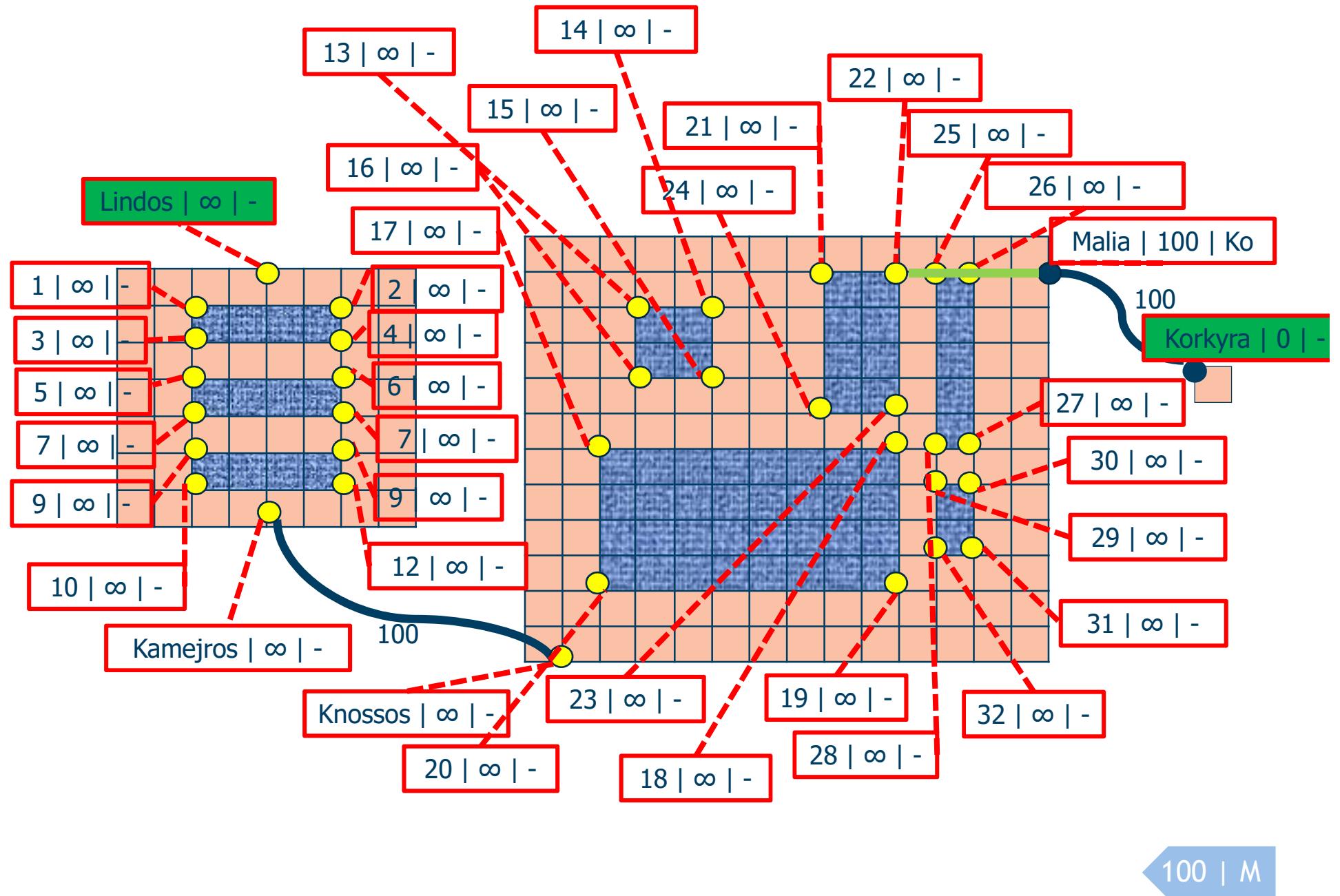


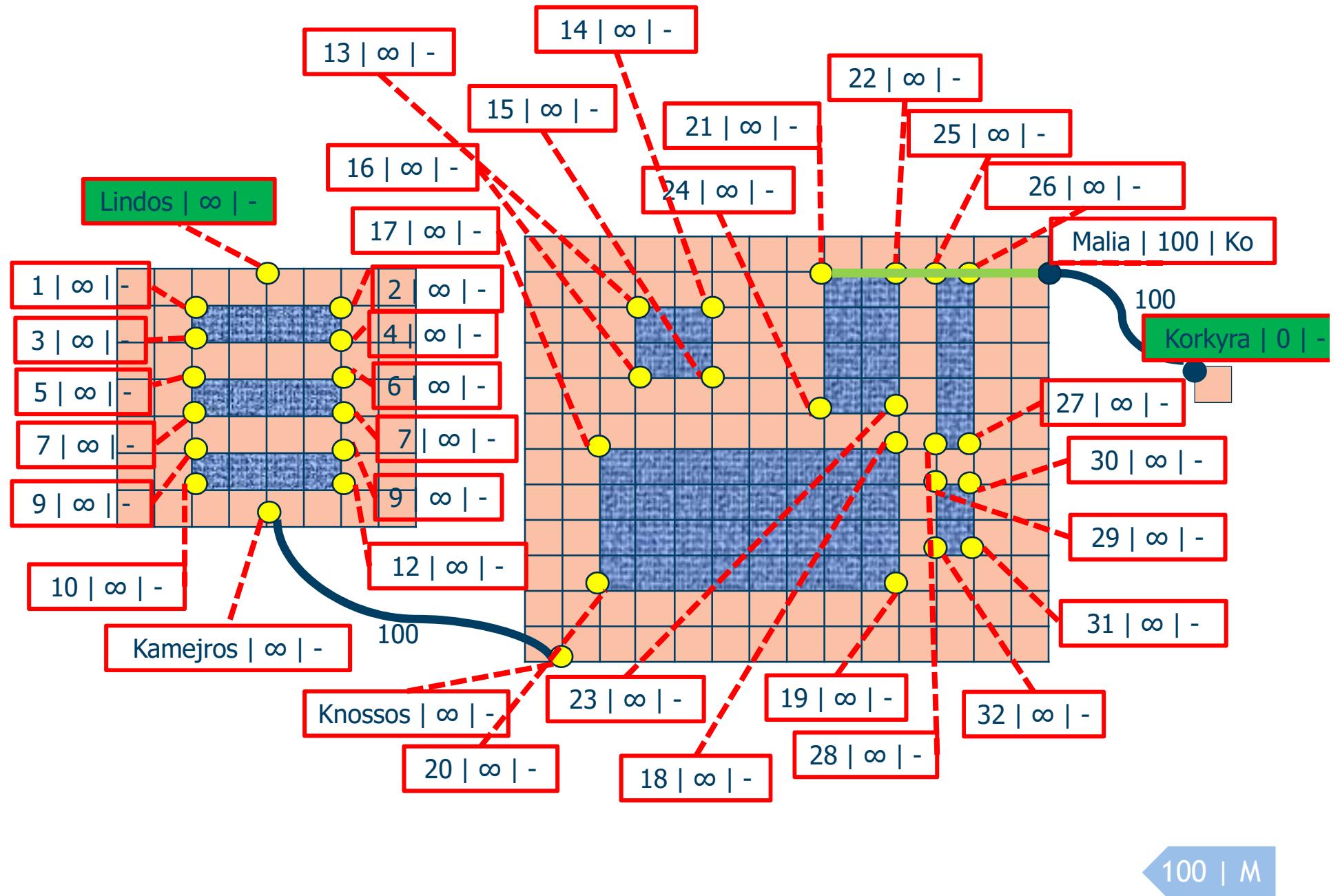


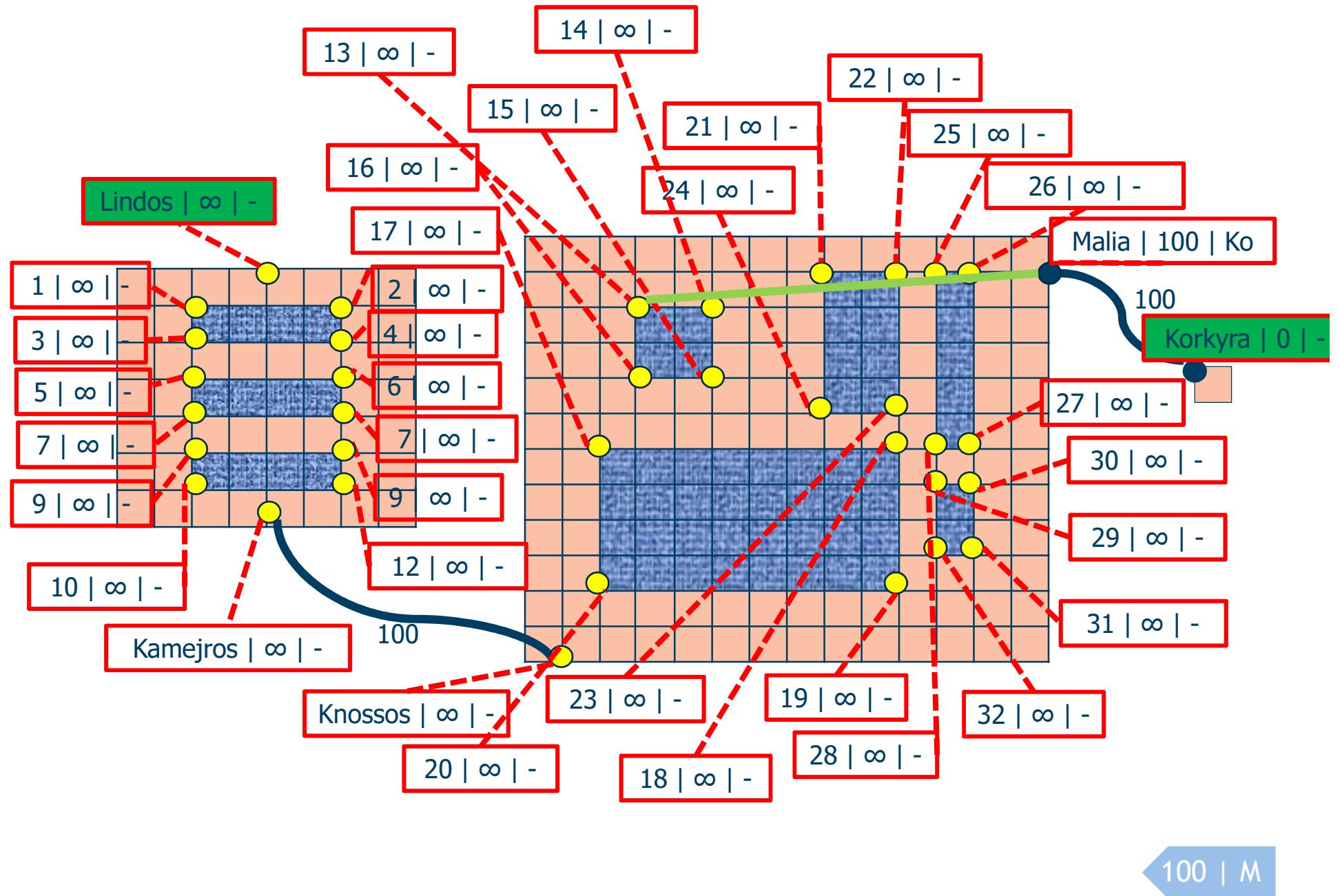


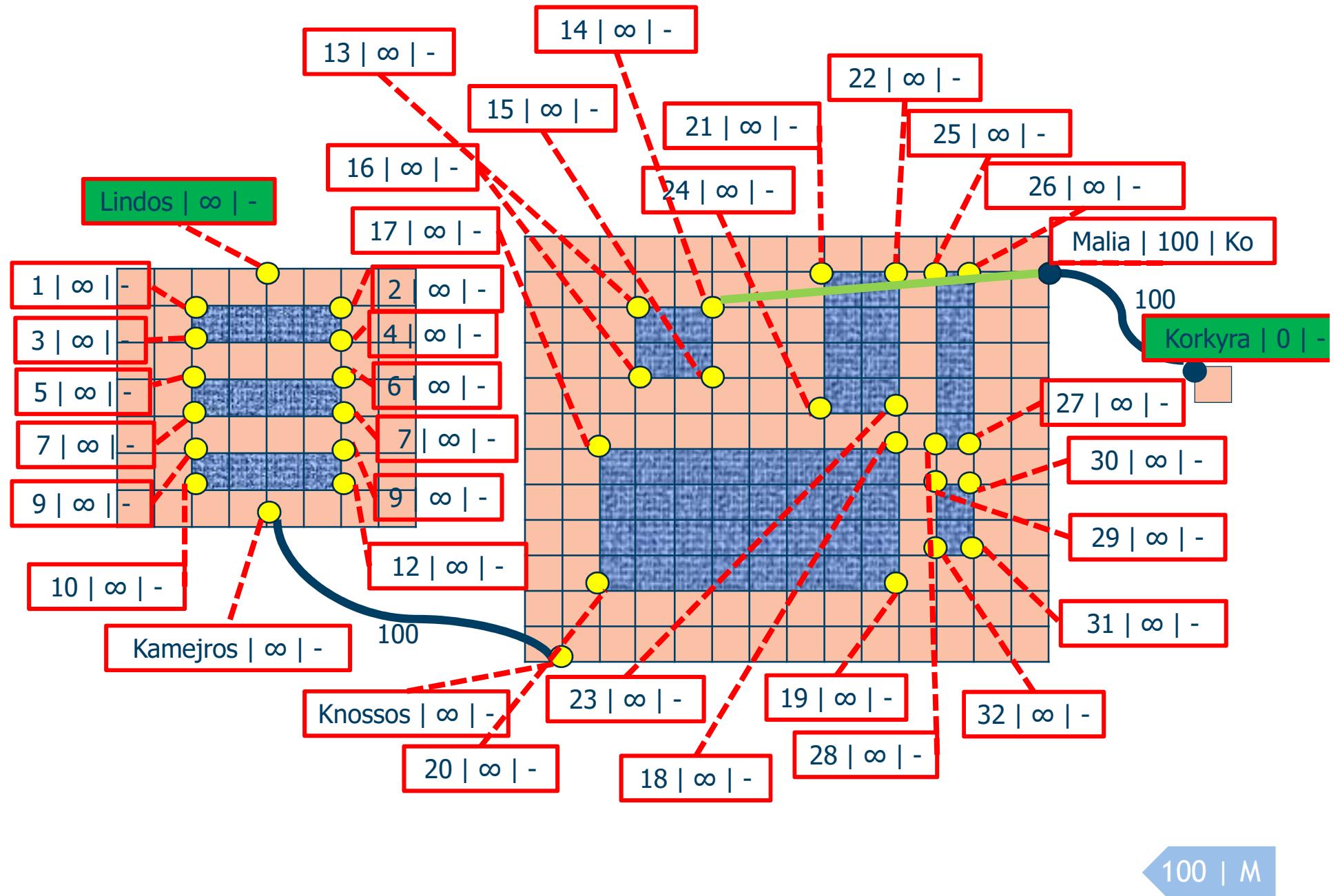


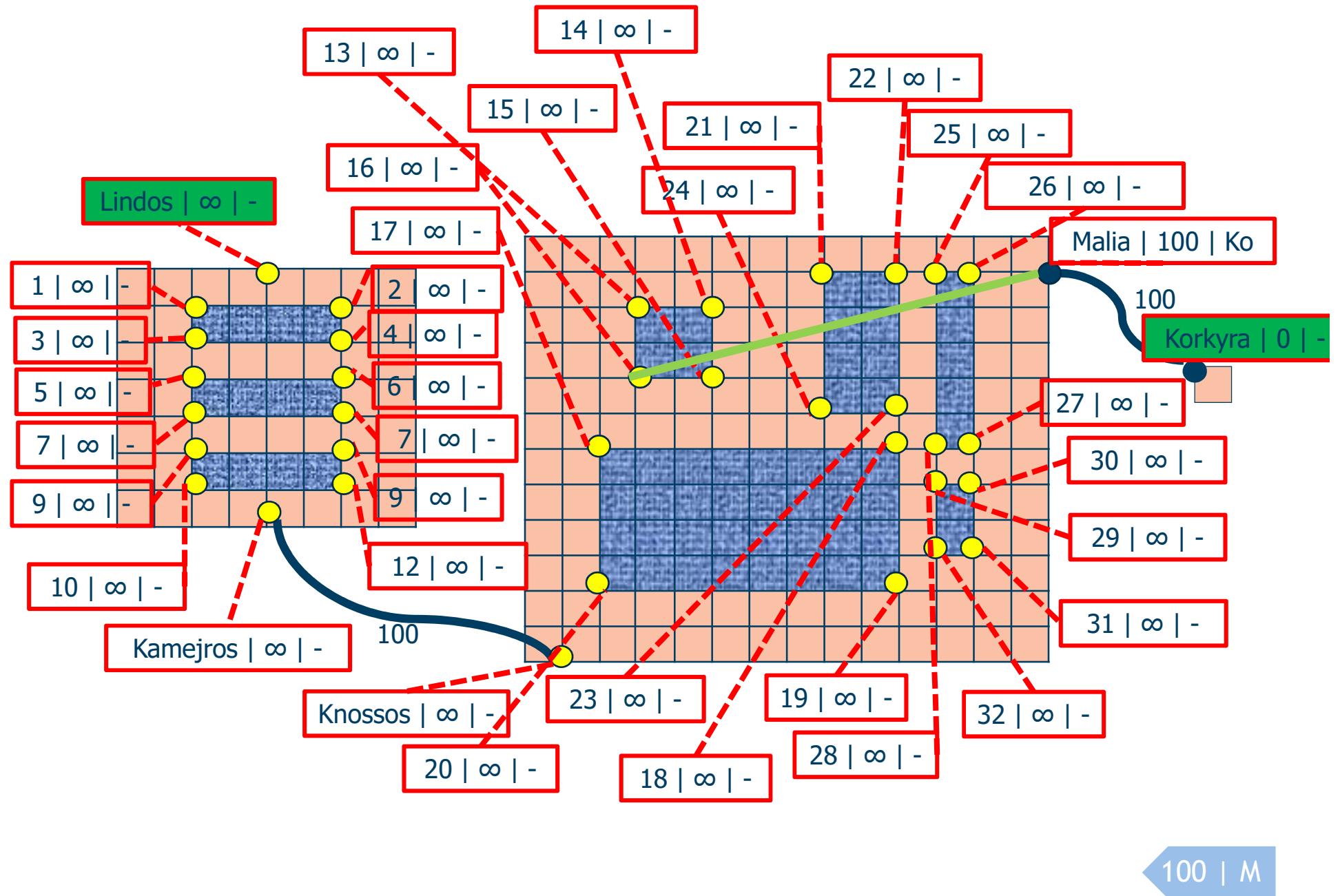


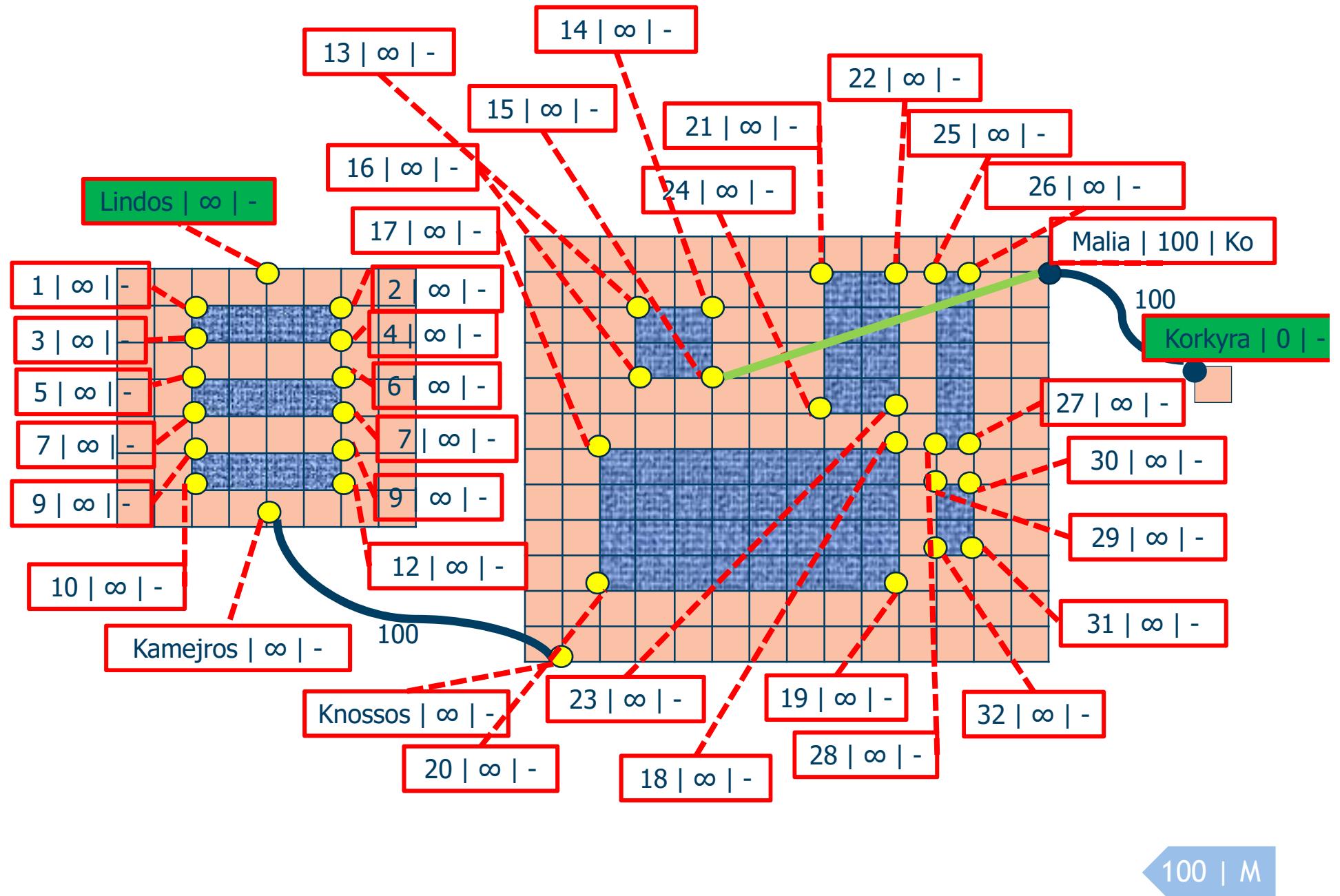


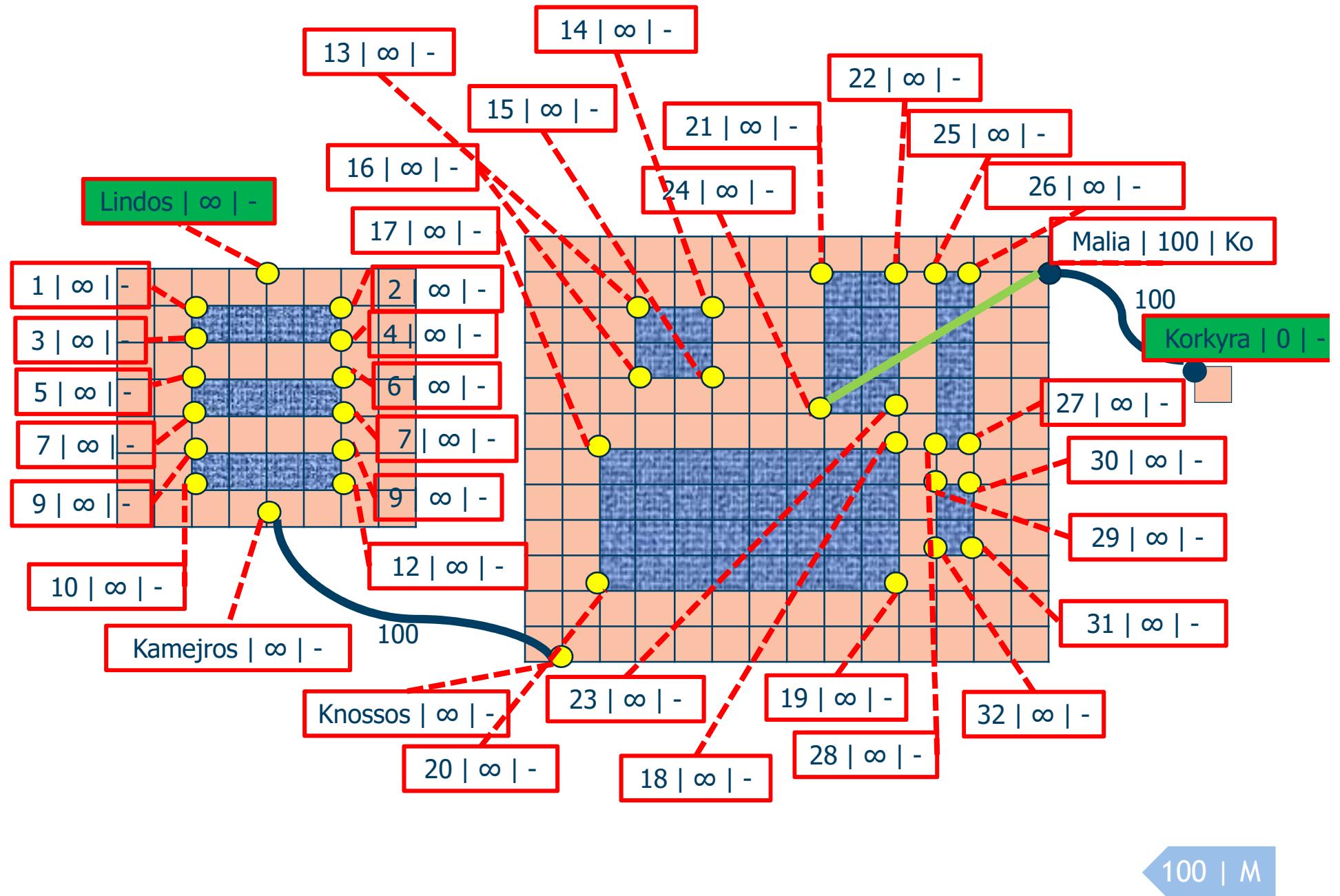


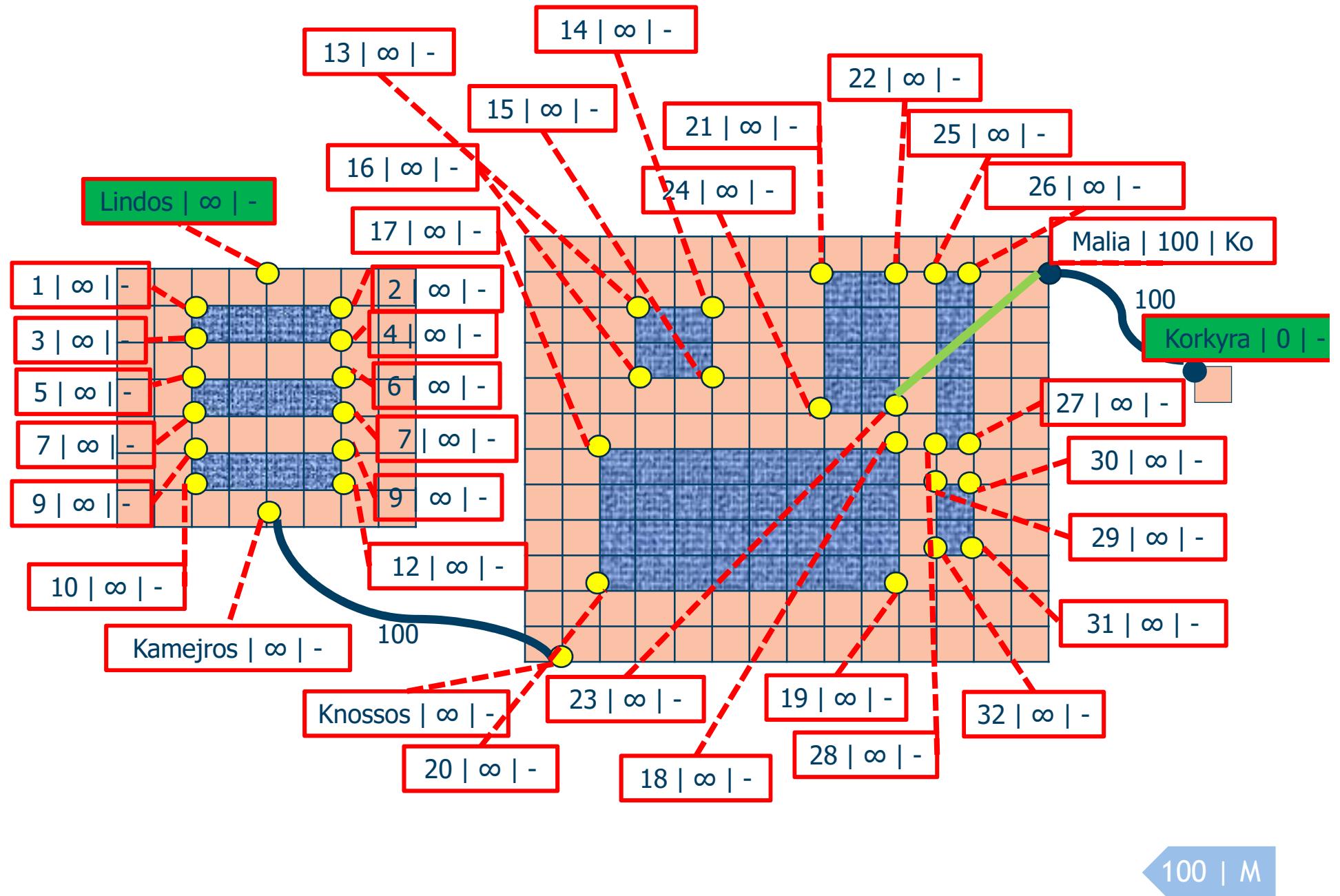


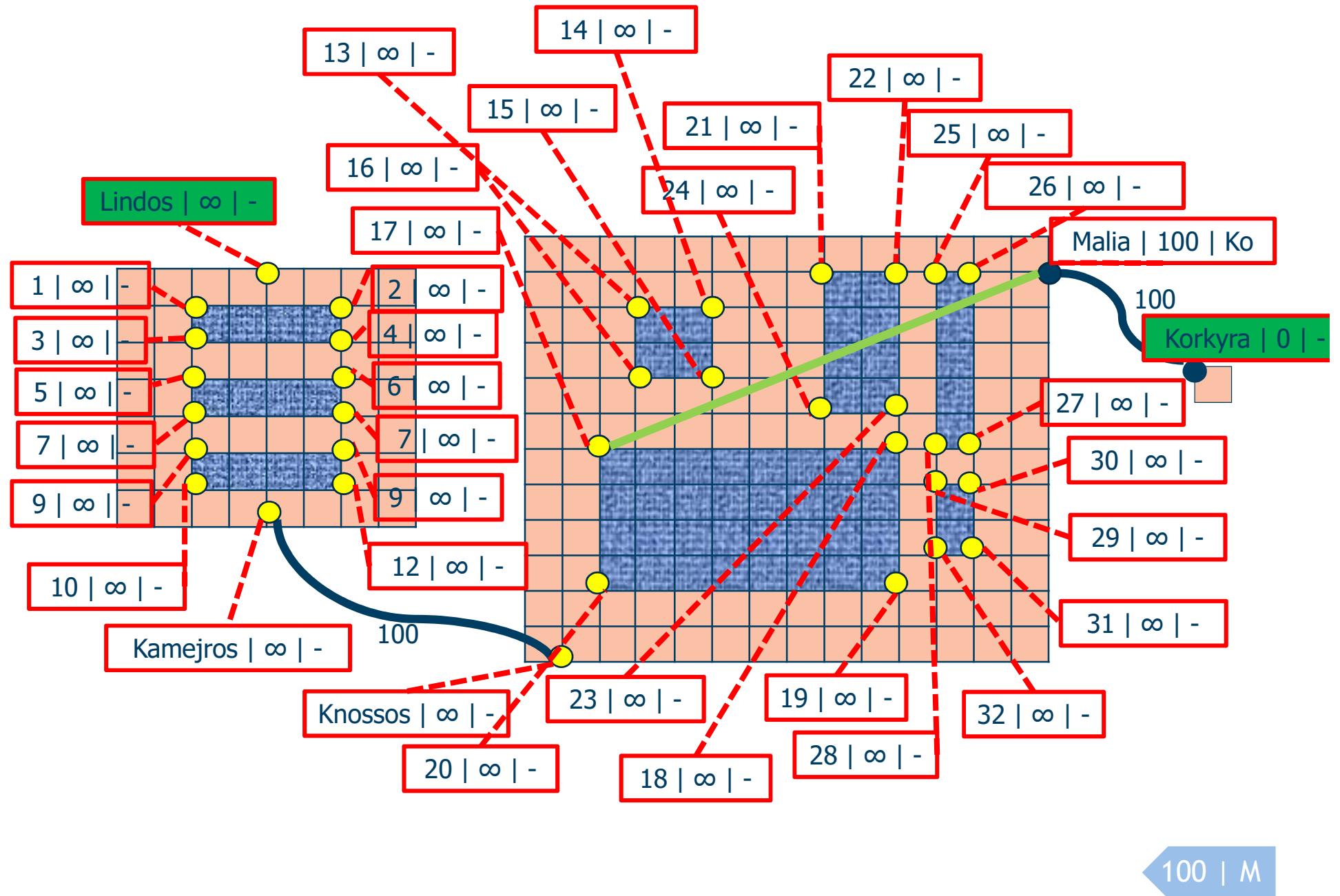


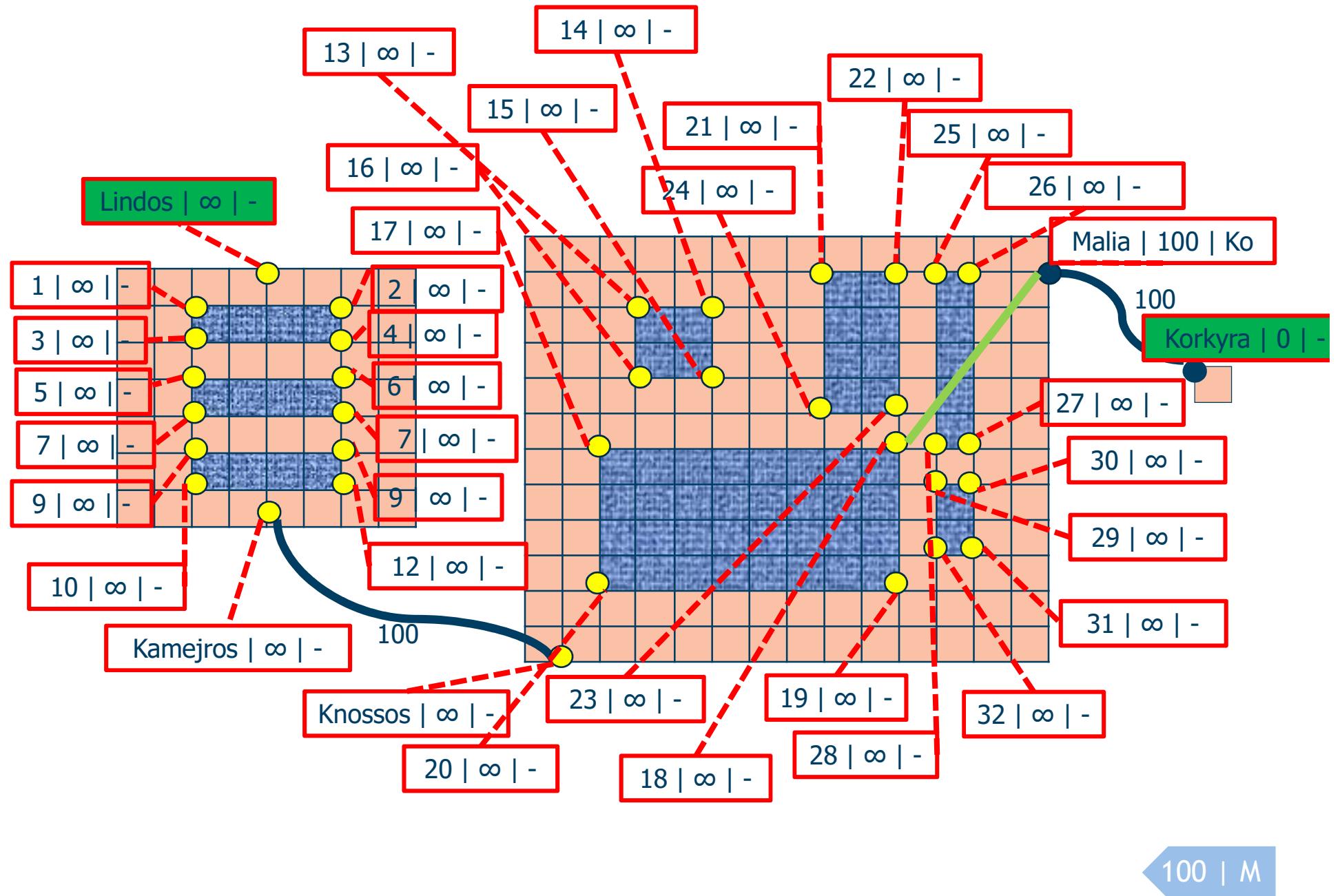


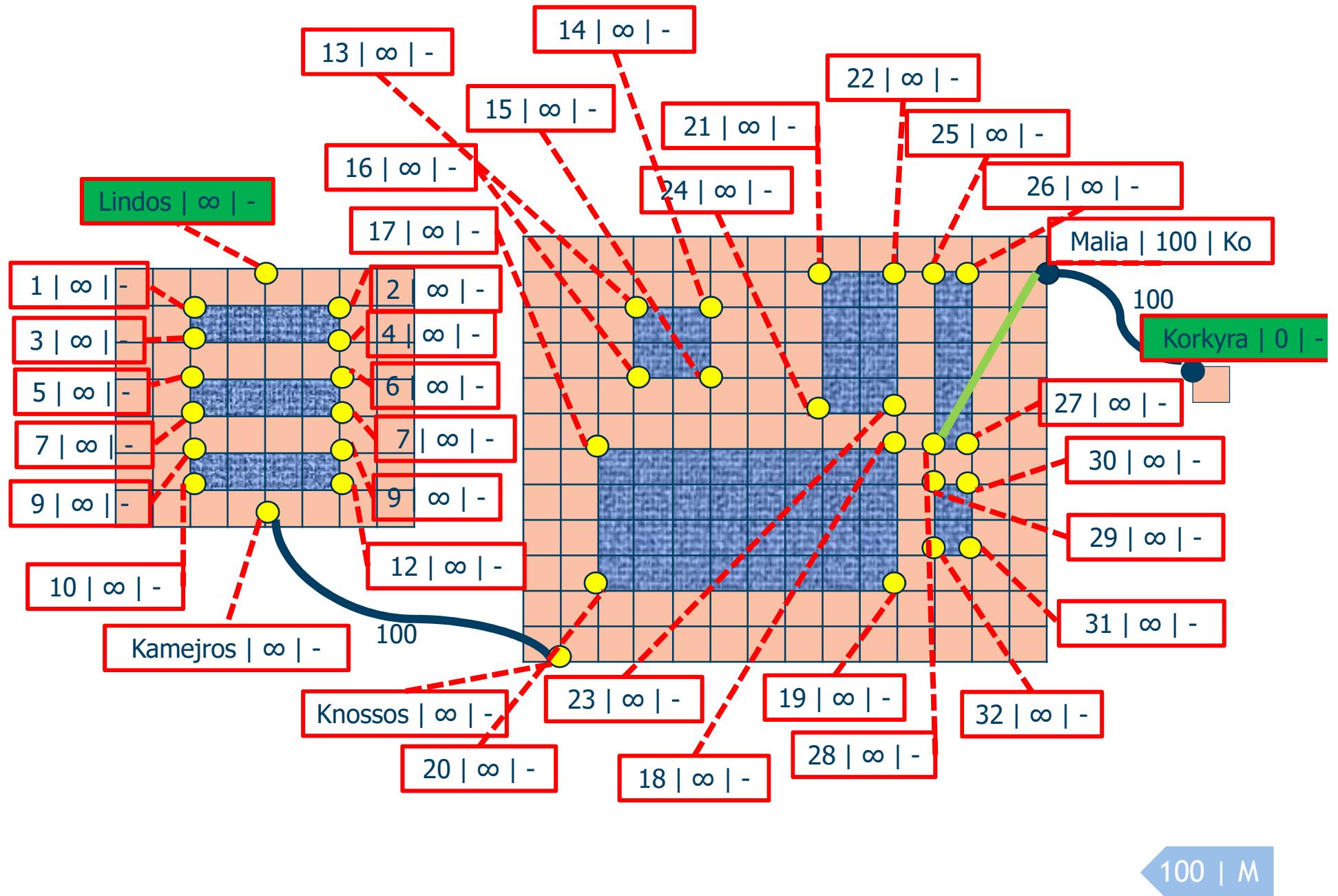


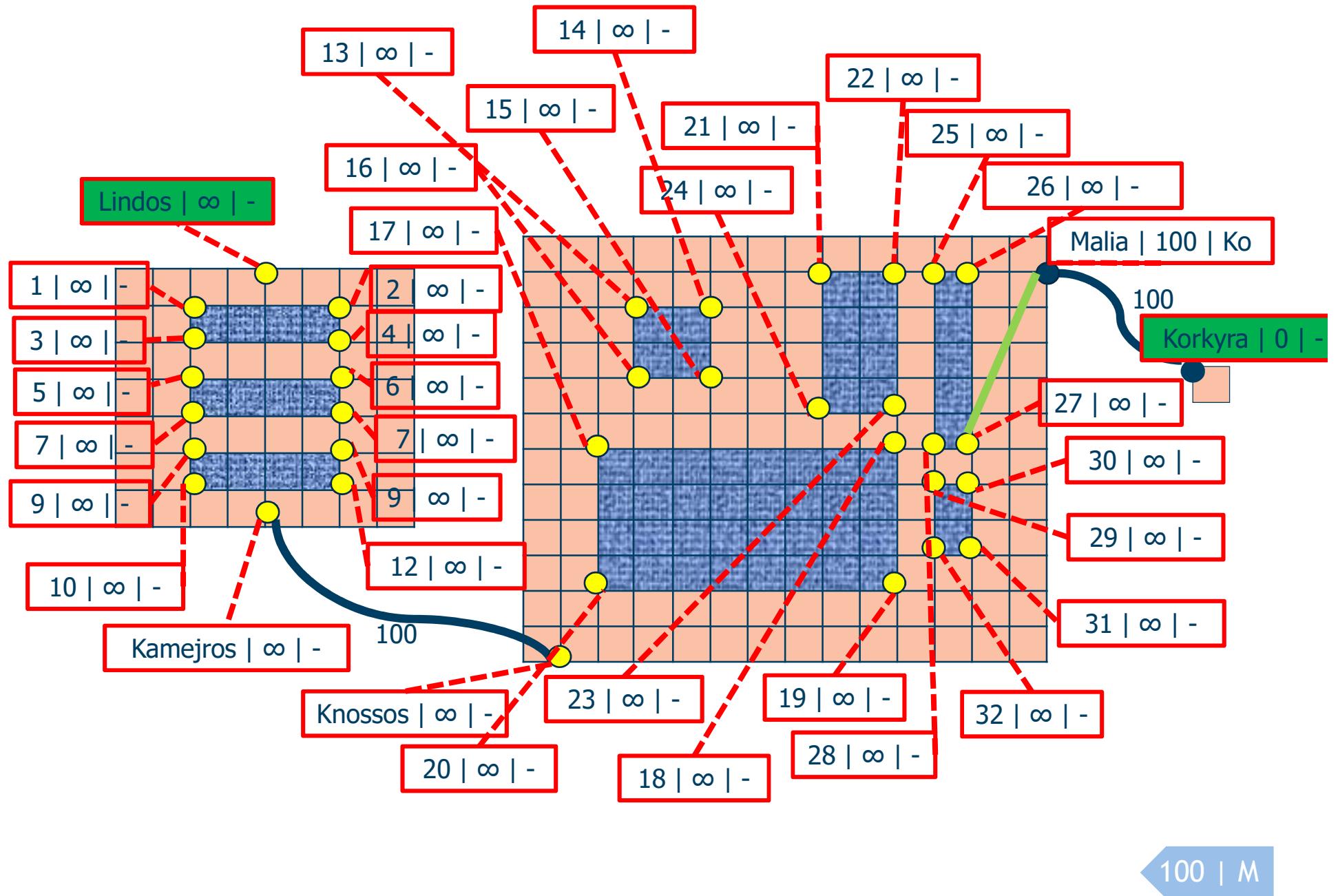


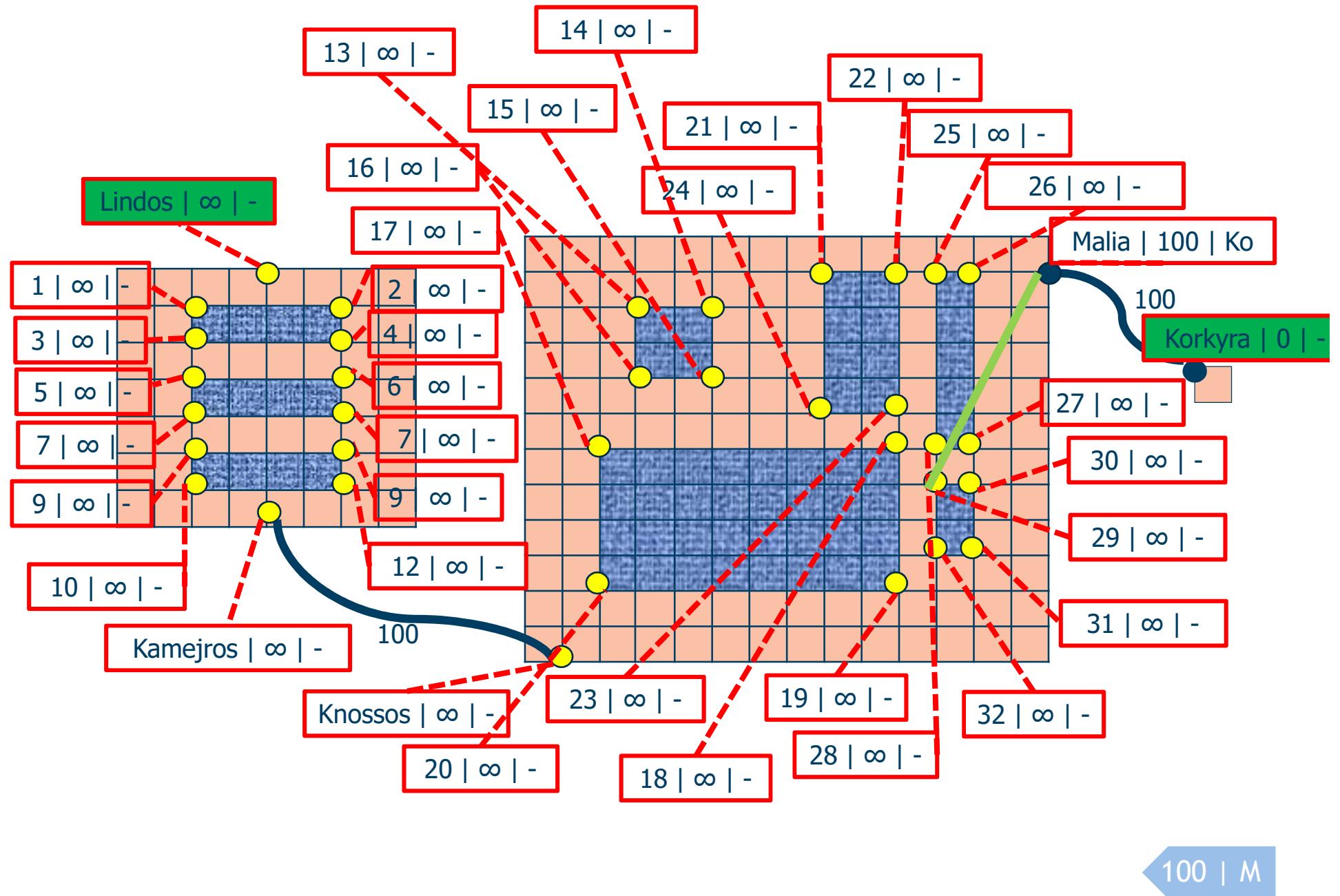


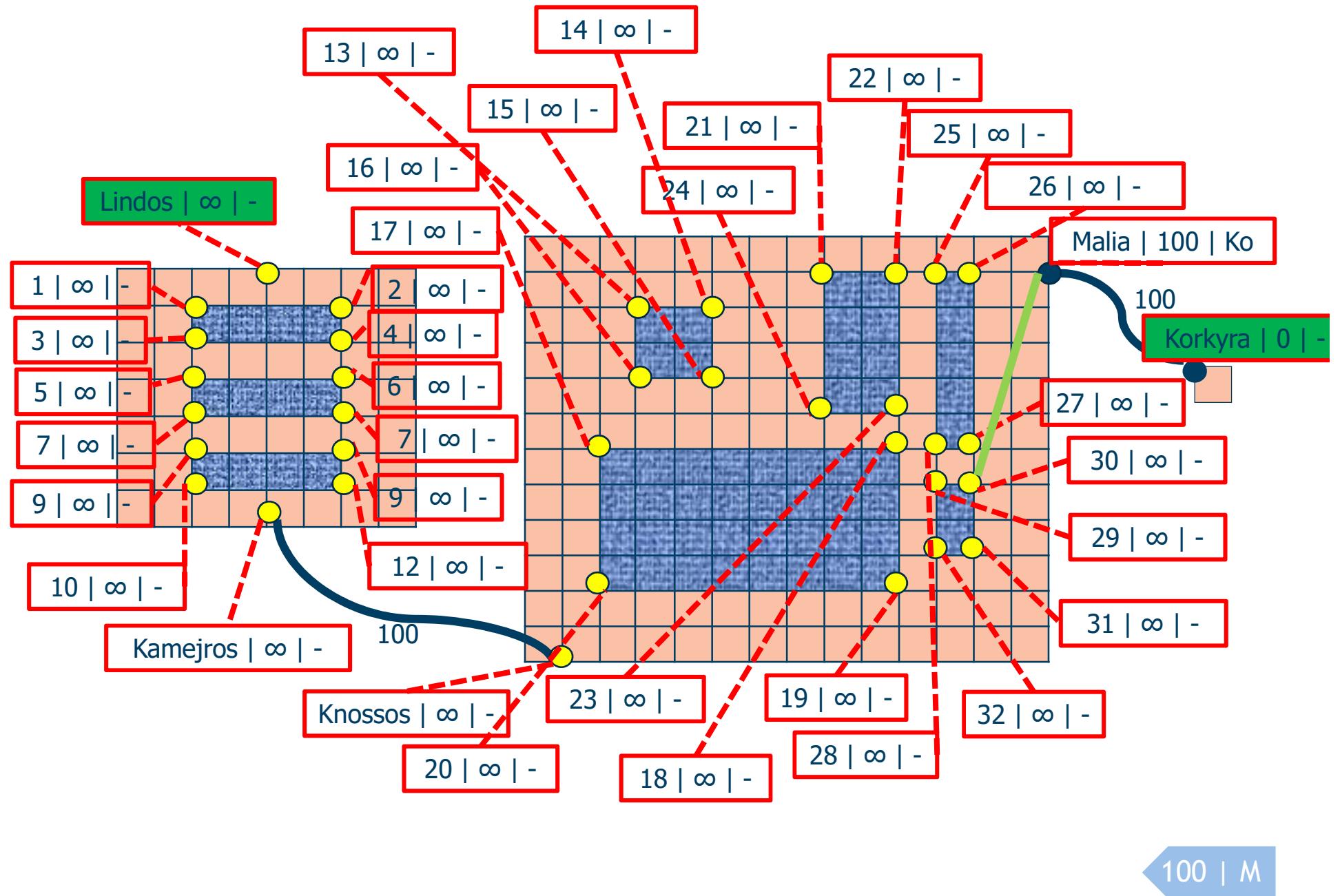


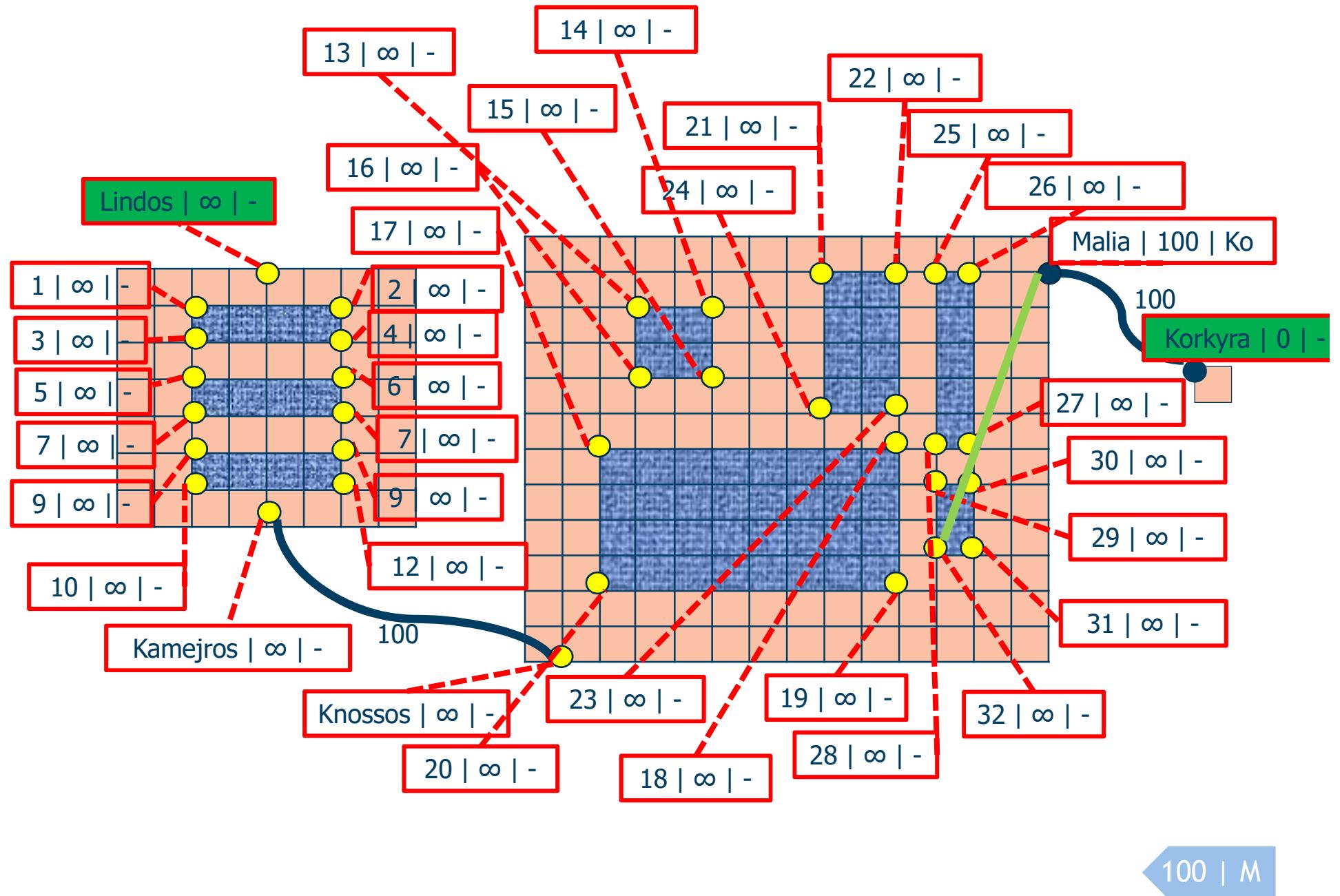


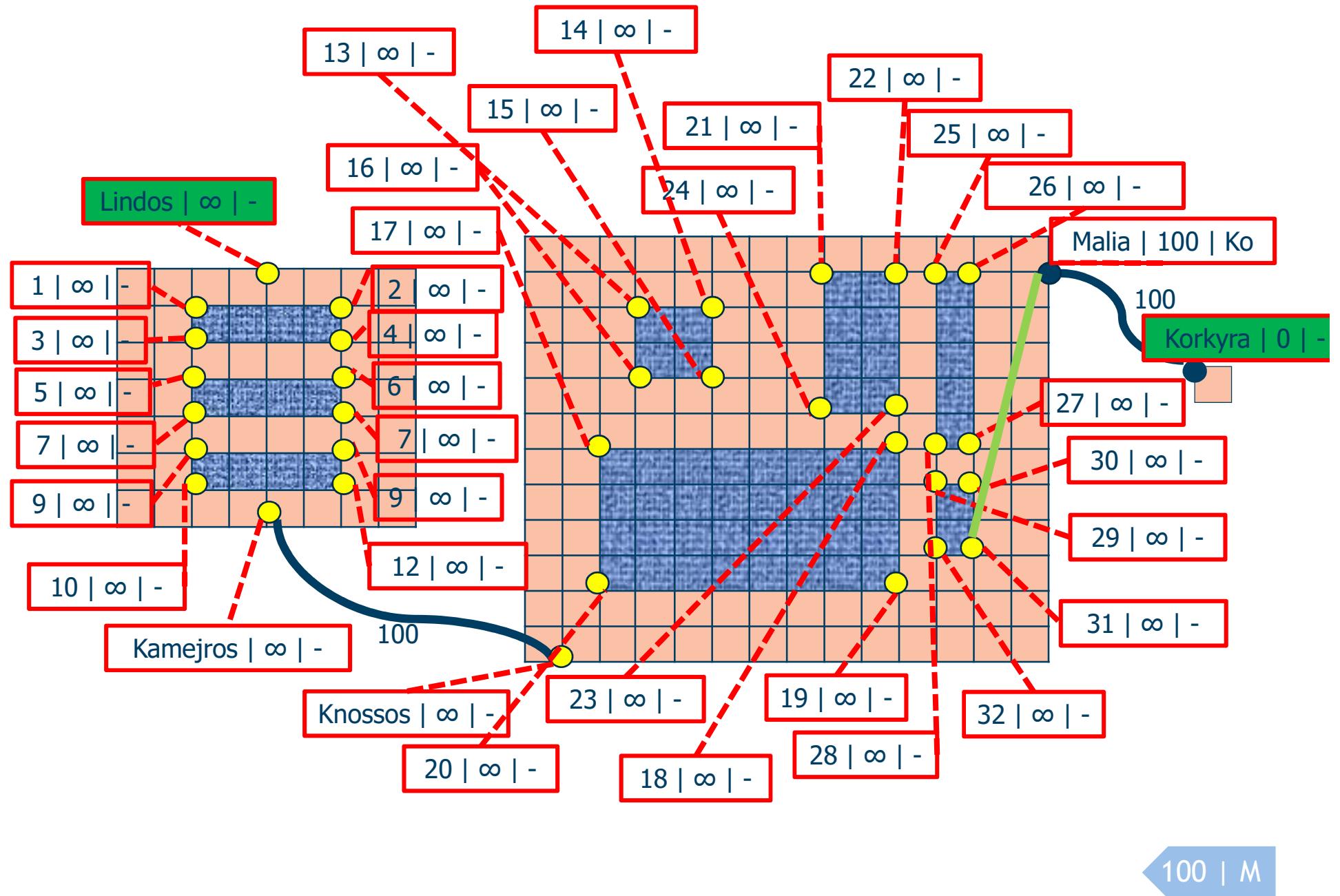


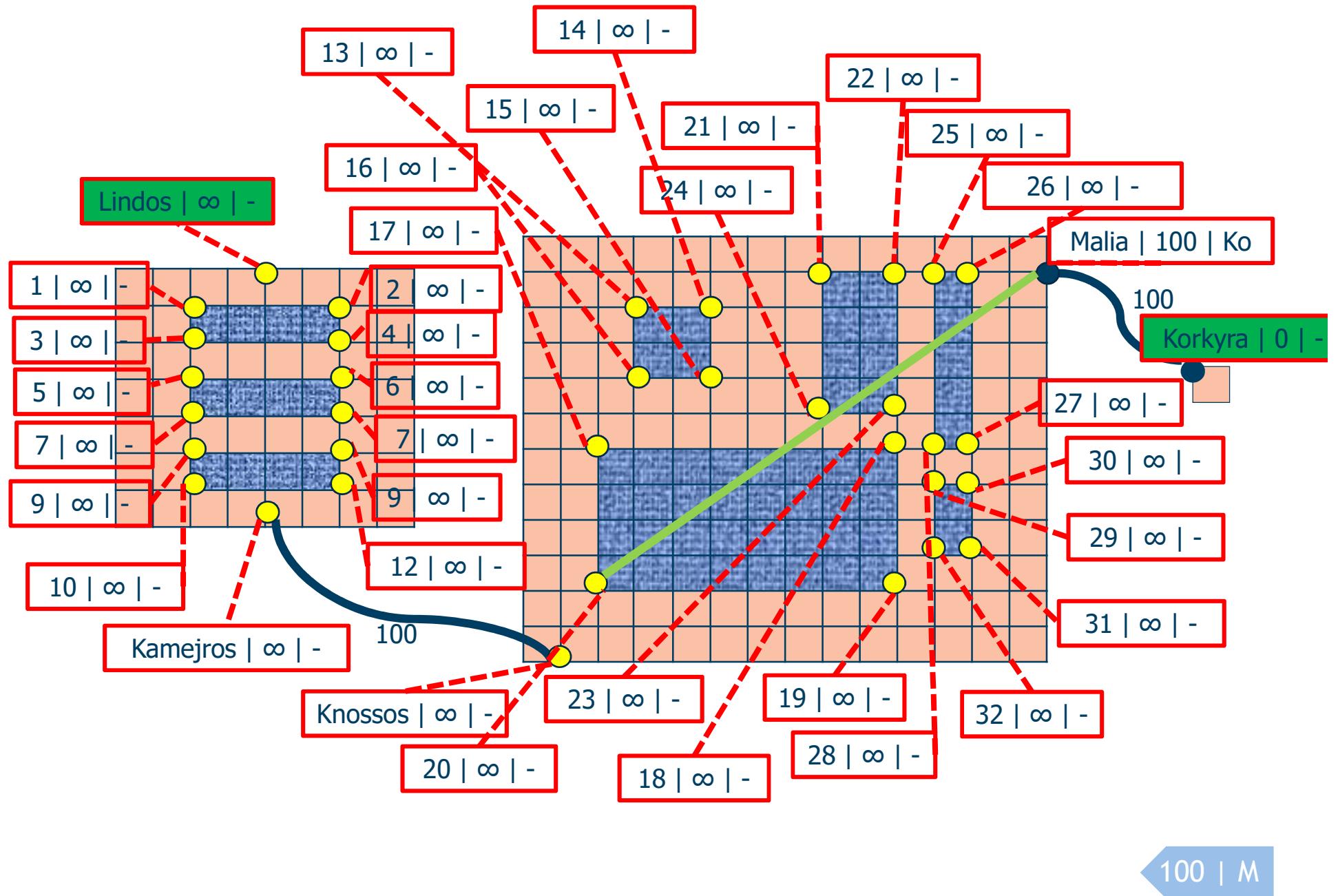


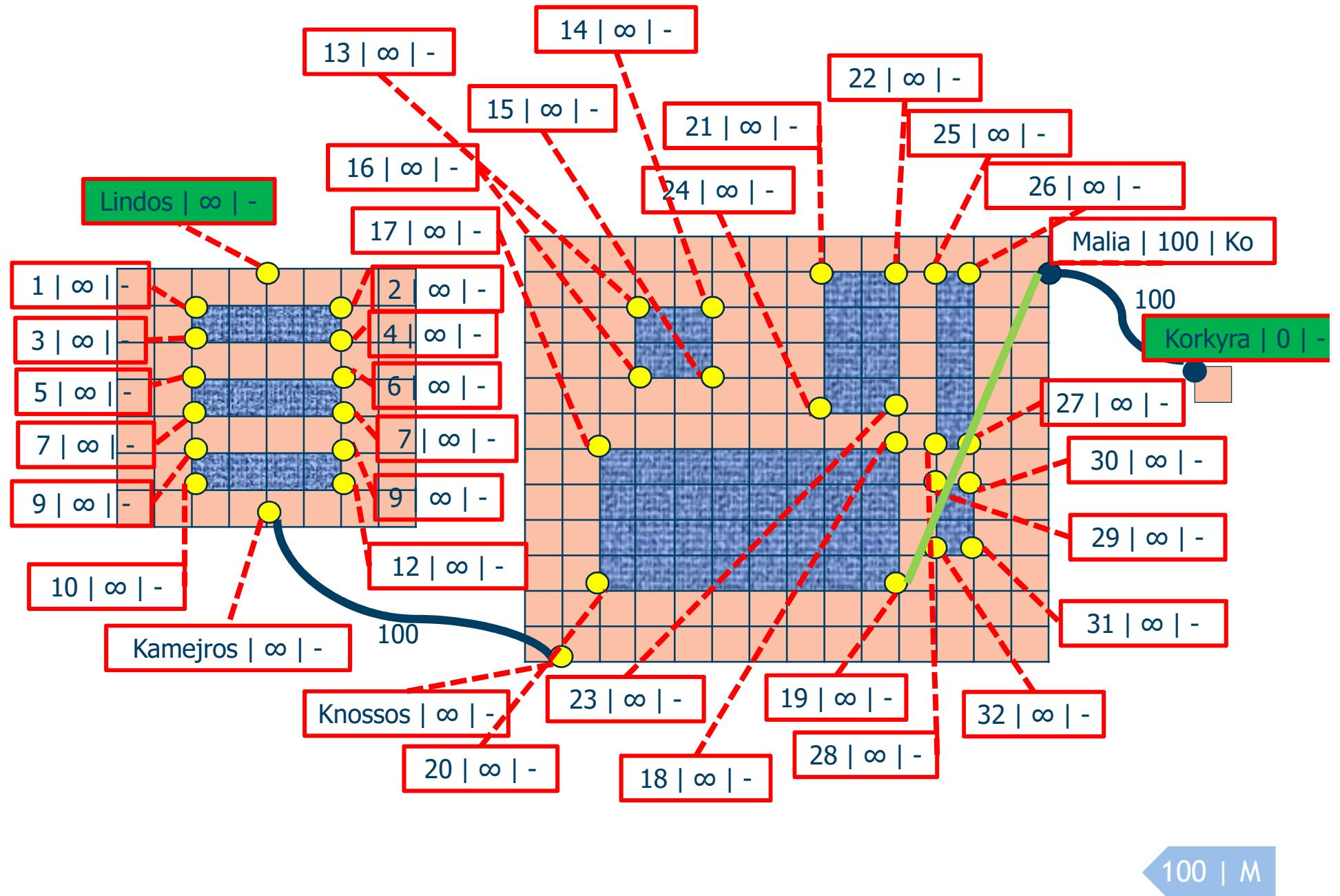


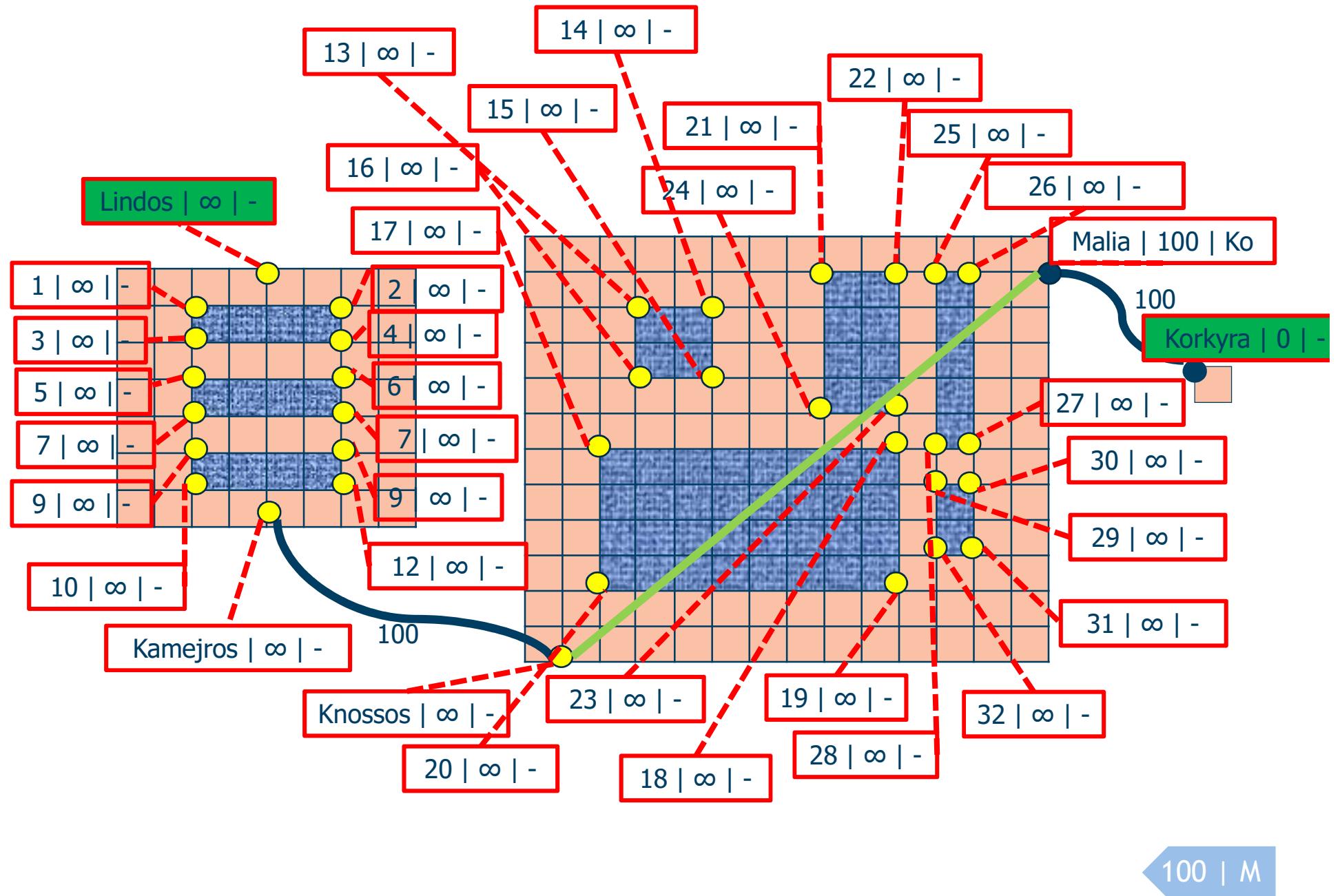


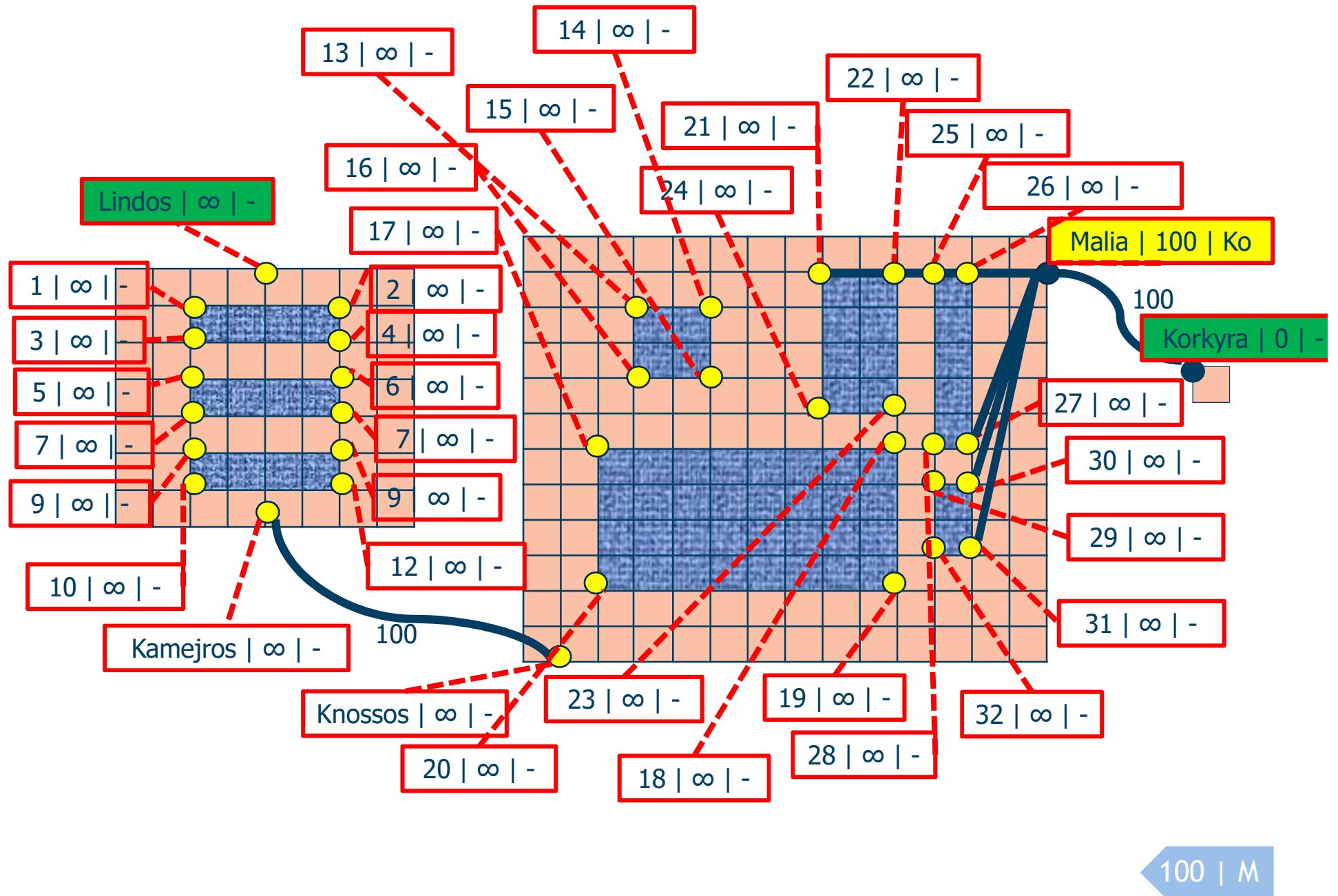


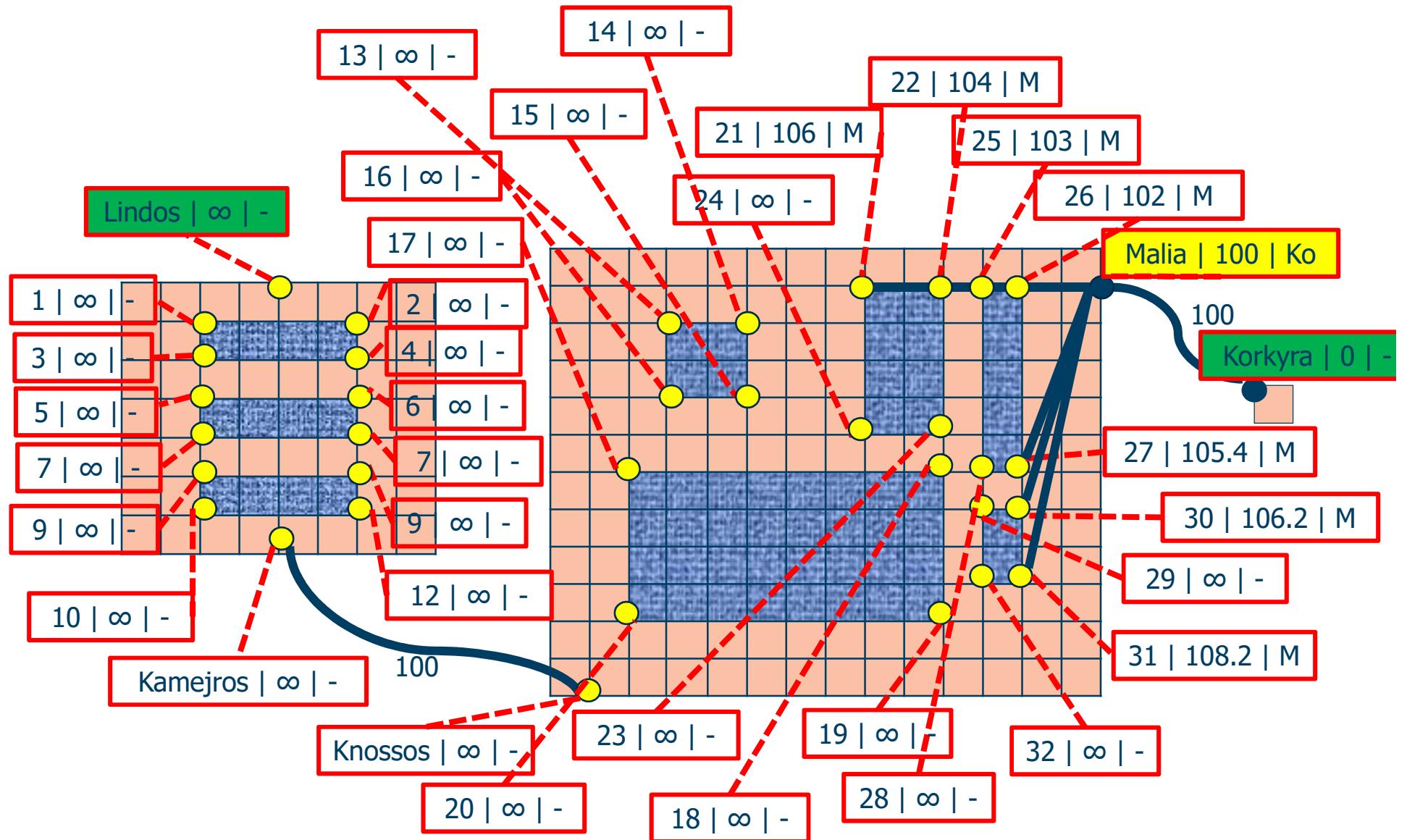












102 | 26
103 | 25
10 June 2017

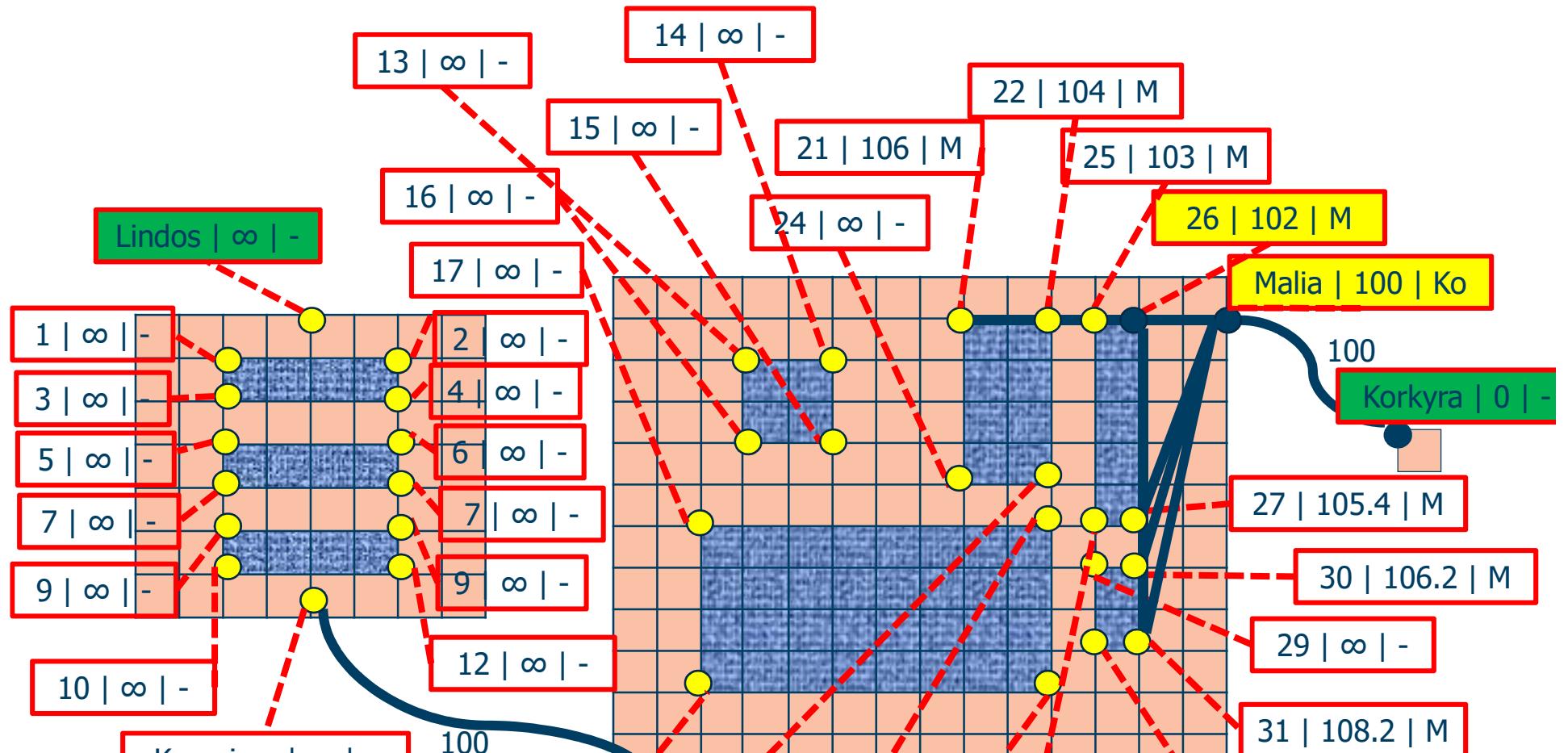
104 | 22

105.4 | 27
Tugas Akhir - KI1502

106 | 21

106.2 | 30

108.2 | 31
100



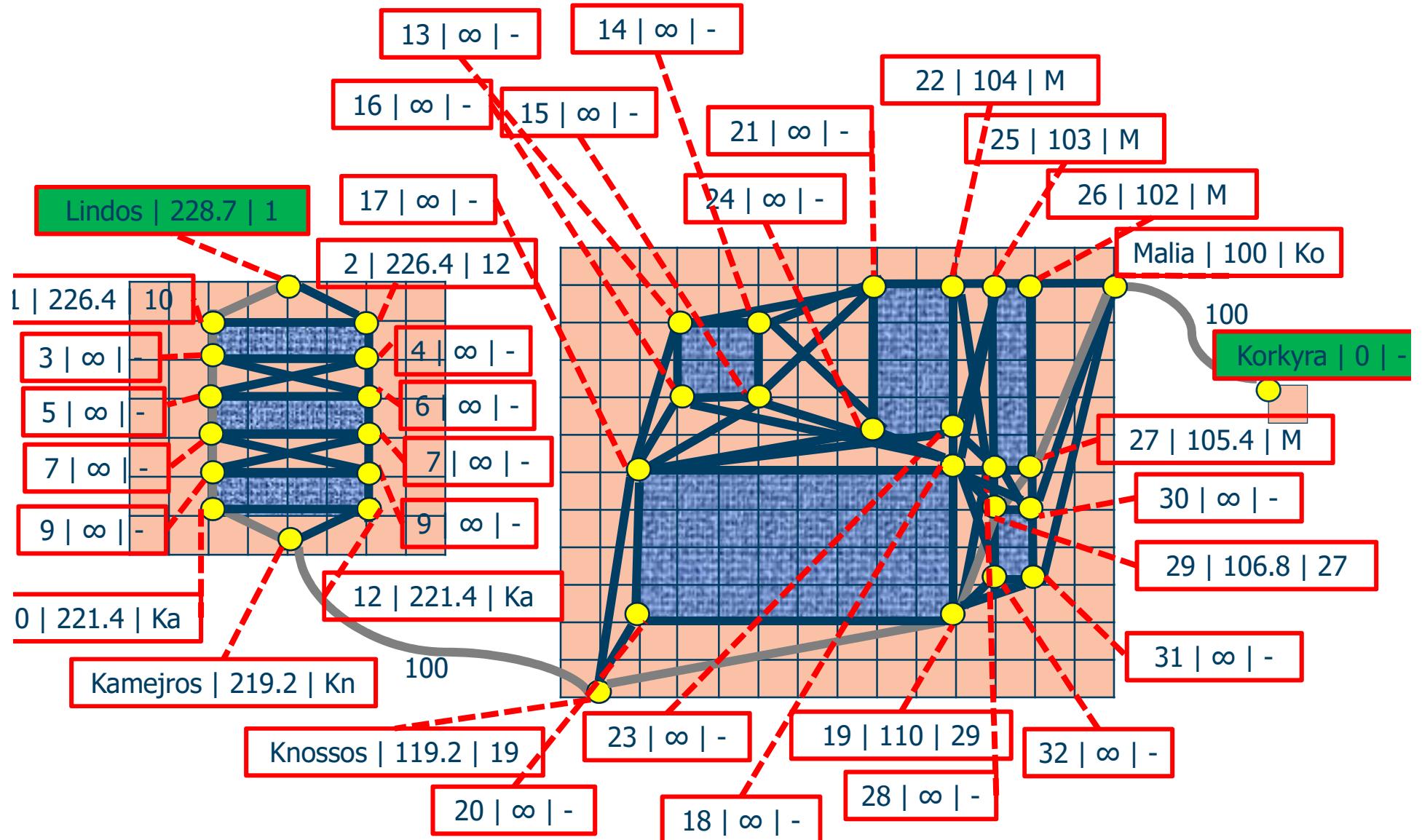
103 | 25
10 June 2017

104 | 22

105.4 | 27
106 | 21
Tugas Akhir - KI1502

106.2 | 30

108.2 | 31
101





ITS

Institut
Teknologi
Sepuluh Nopember

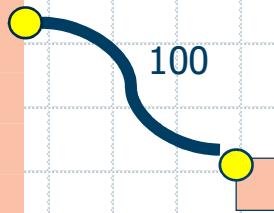
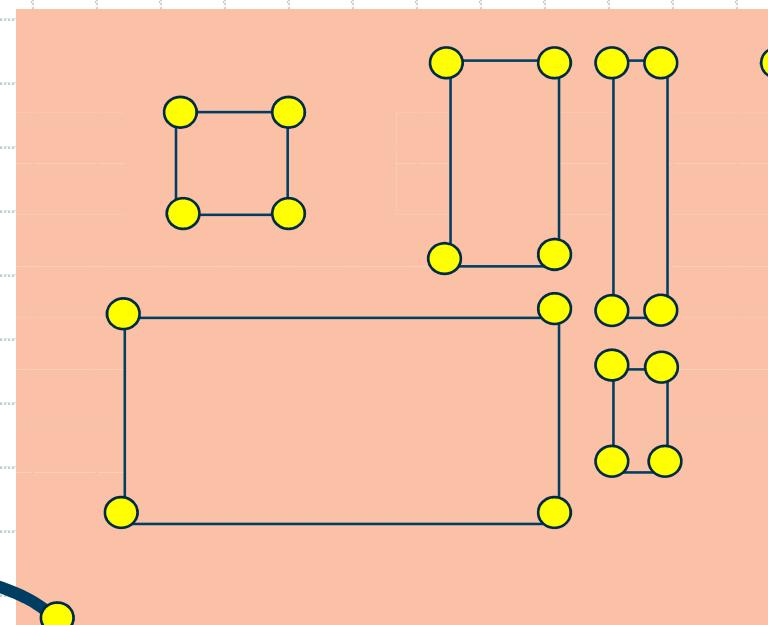
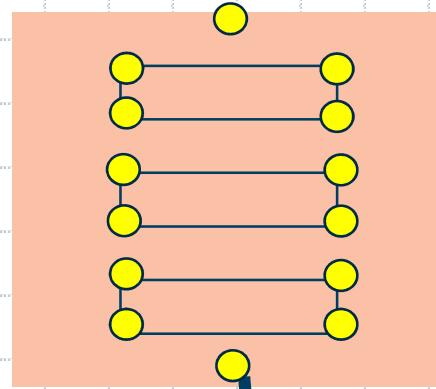
STRATEGI PENYELESAIAN

2. Penggambaran Entitas Halangan

- Halangan berbentuk persegi panjang
- Apabila digambarkan dengan 4 garis, maka:



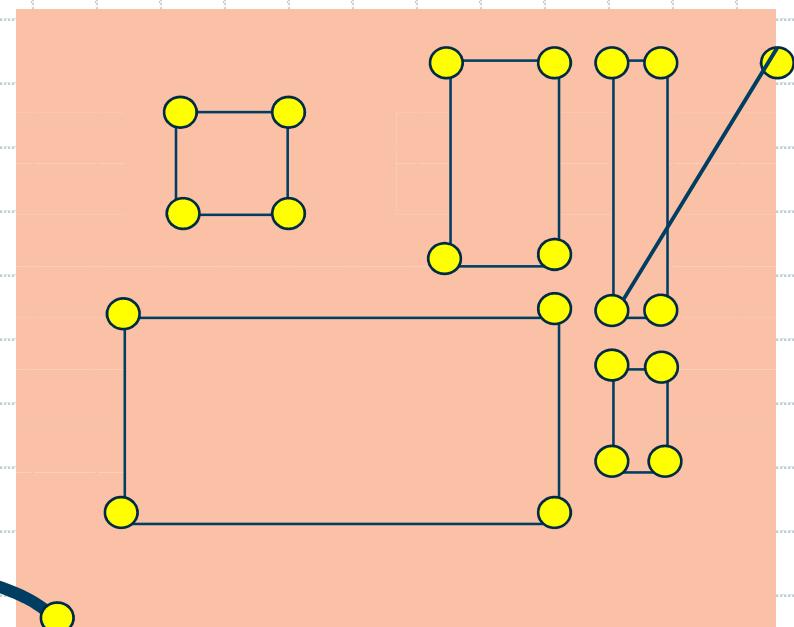
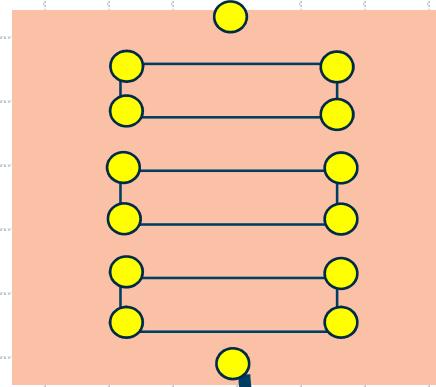
STRATEGI PENYELESAIAN



Ilustrasi Penggambaran Halangan dengan 4 Garis



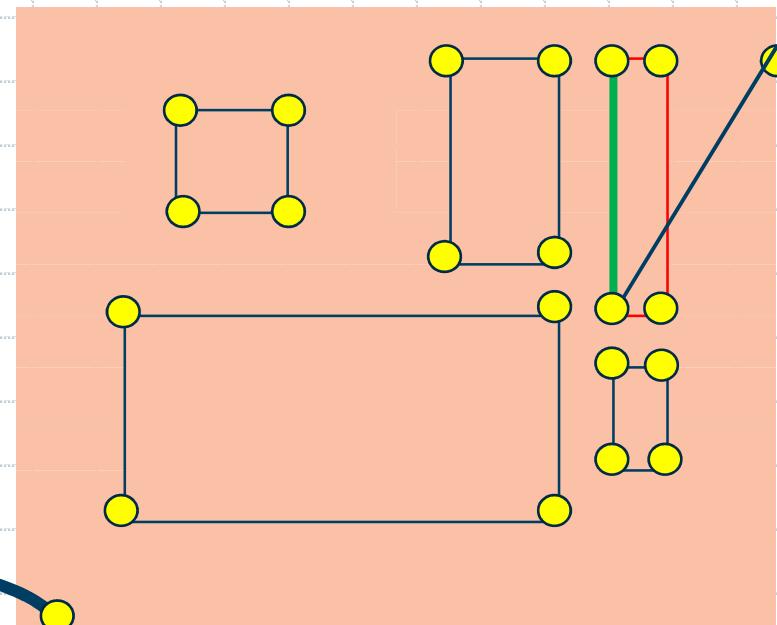
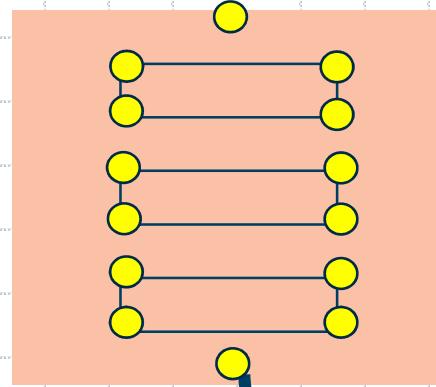
STRATEGI PENYELESAIAN



Pengecekan Visibilitas dengan Tiap Garis Halangan



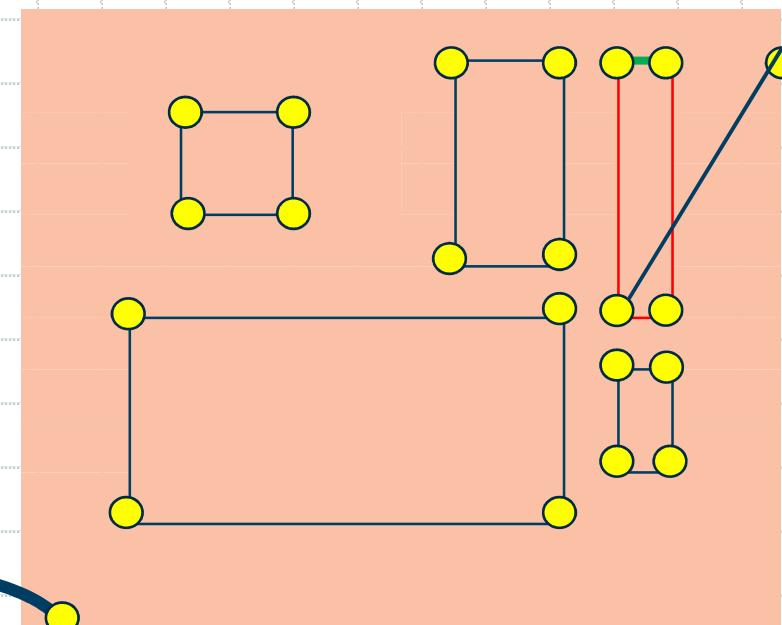
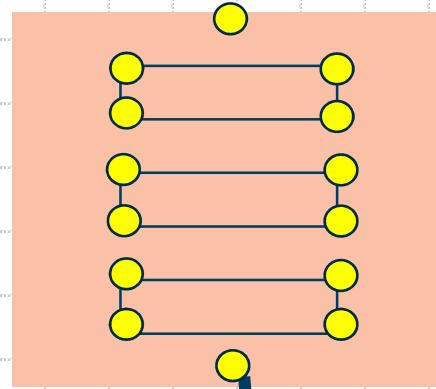
STRATEGI PENYELESAIAN



Pengecekan Visibilitas dengan Tiap Garis Halangan (1)



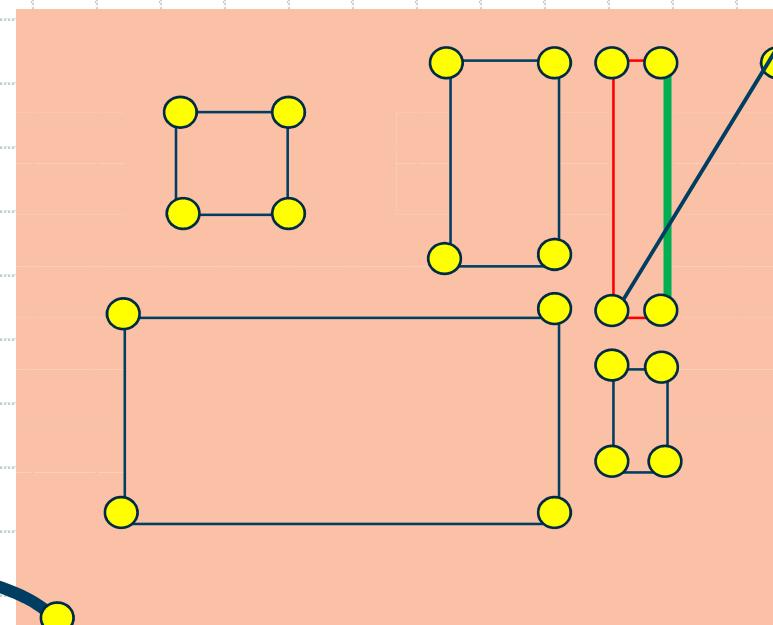
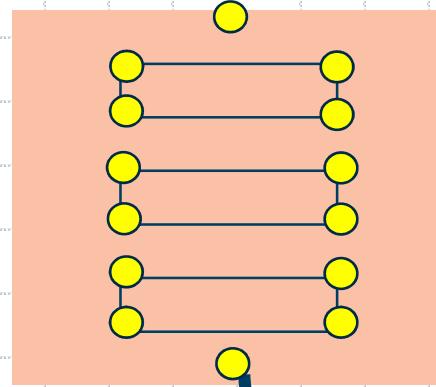
STRATEGI PENYELESAIAN



Pengecekan Visibilitas dengan Tiap Garis Halangan (2)

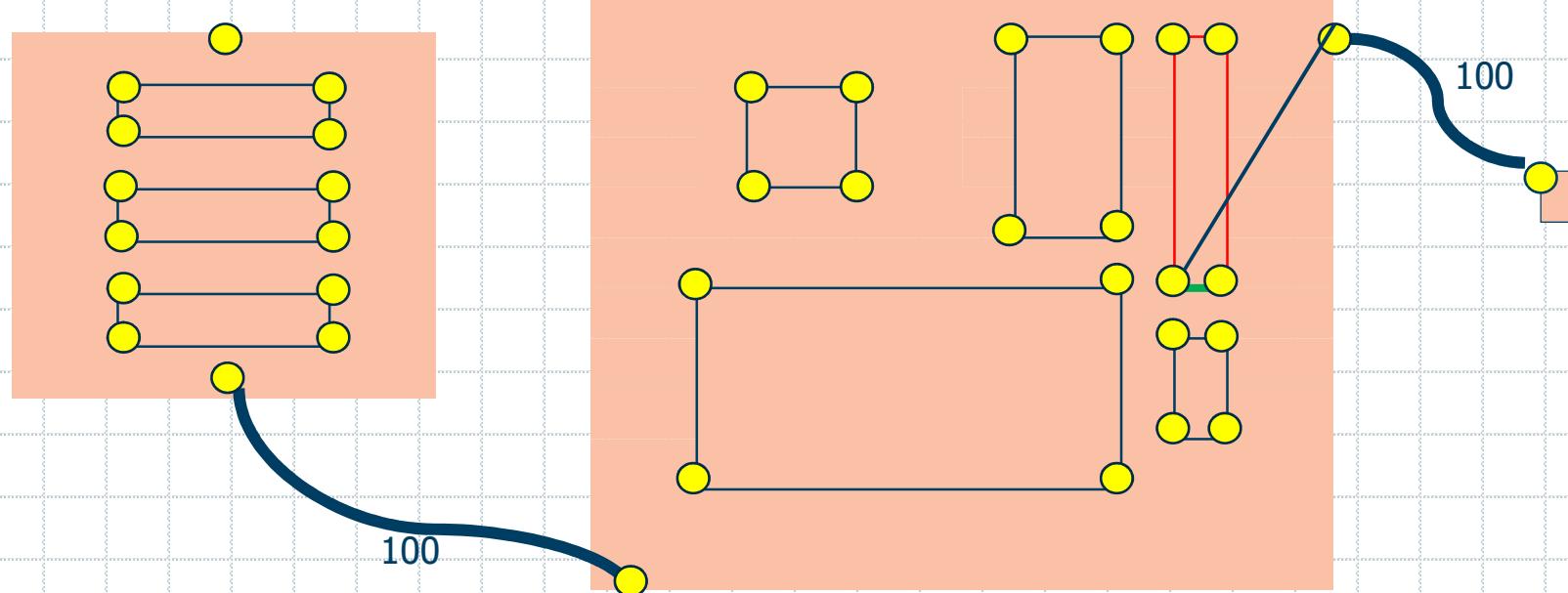


STRATEGI PENYELESAIAN



Pengecekan Visibilitas dengan Tiap Garis Halangan (3)

STRATEGI PENYELESAIAN



Pengecekan Visibilitas dengan Tiap Garis Halangan (4)



ITS

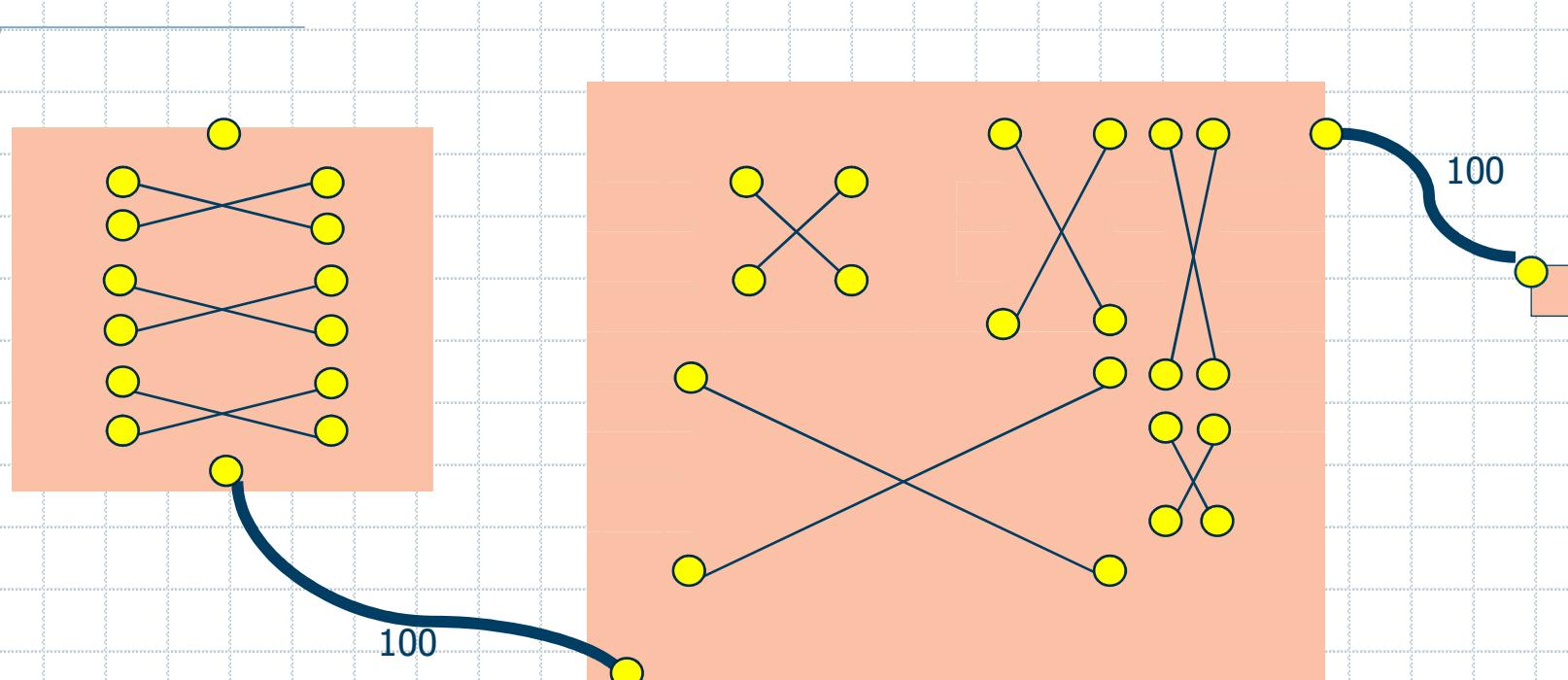
Institut
Teknologi
Sepuluh Nopember

STRATEGI PENYELESAIAN

2. Penggambaran Entitas Halangan

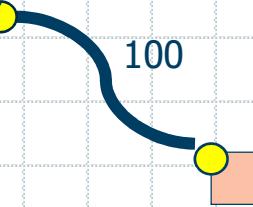
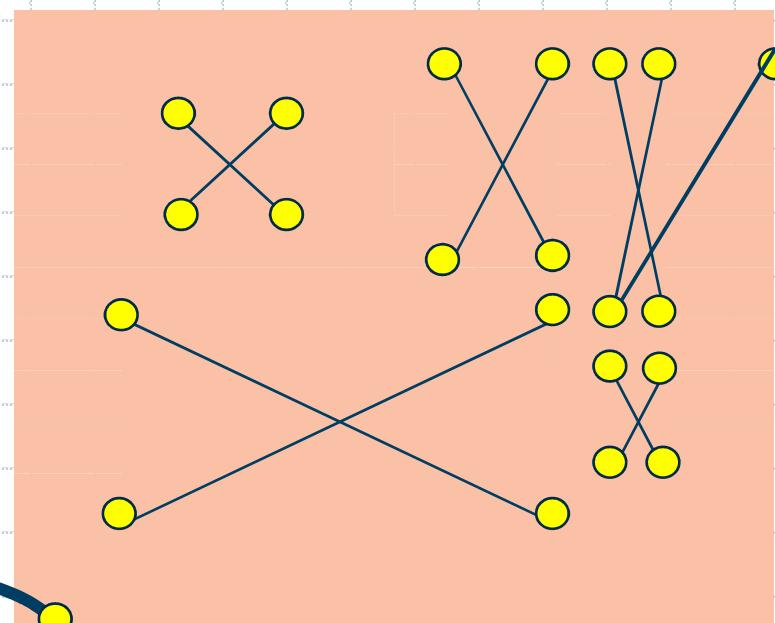
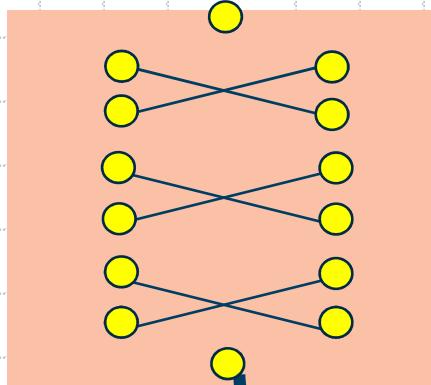
- Sebenarnya, halangan juga dapat digambarkan dengan 2 garis diagonal

STRATEGI PENYELESAIAN



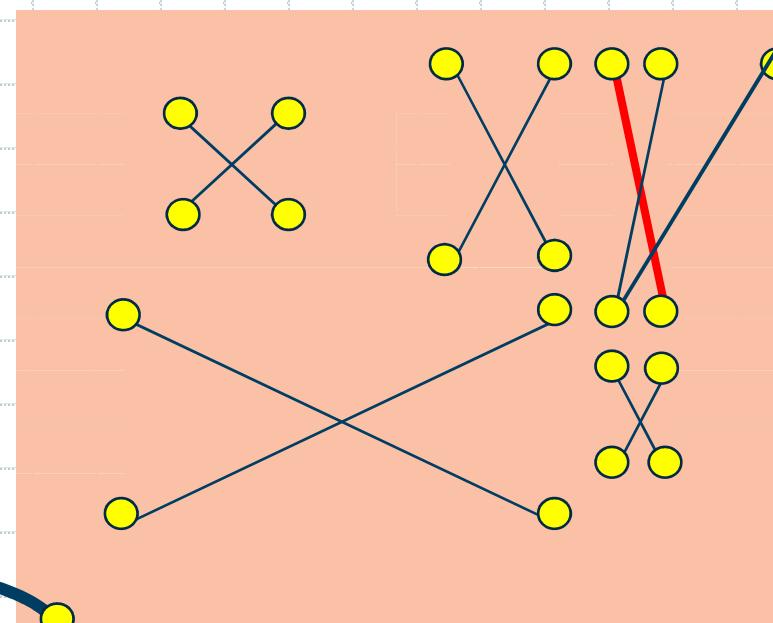
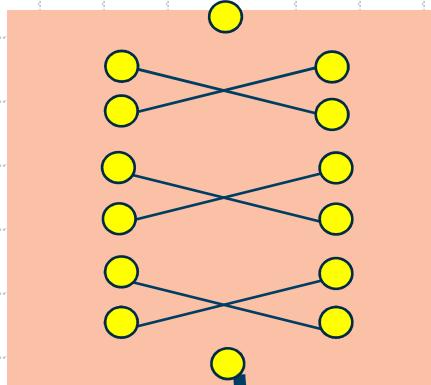


STRATEGI PENYELESAIAN





STRATEGI PENYELESAIAN

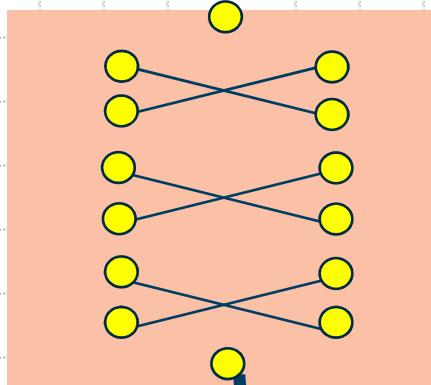


100

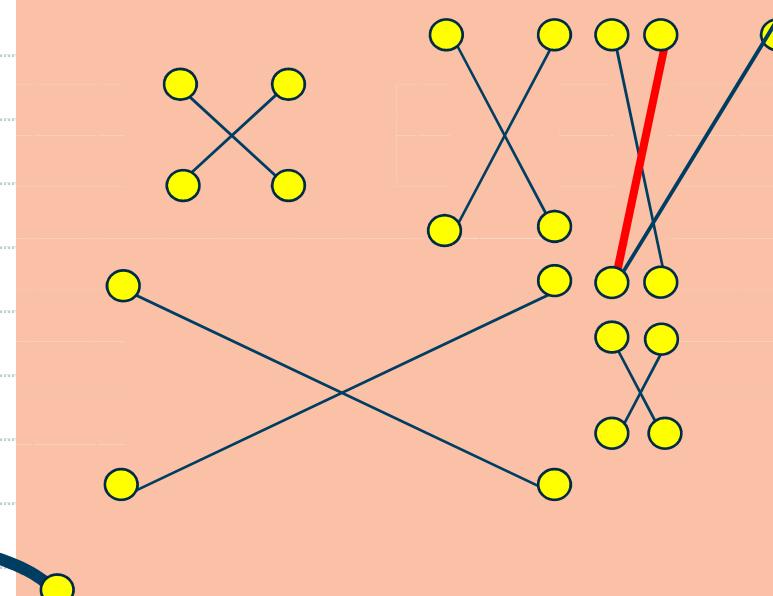
113



STRATEGI PENYELESAIAN



100



100





ITS

Institut
Teknologi
Sepuluh Nopember

STRATEGI PENYELESAIAN

3. Menghitung Total Jarak Rute dan Penyimpanan Posisi

- Menggunakan nilai jarak dan *vertex* asal (*previous*) dari tiap *vertex*
- Iterasi dari *vertex* tujuan hingga *vertex* sumber
- Pada tiap iterasi,
 - Jika *vertex* sekarang dan *vertex* asal berada di satu pulau
 - ◆ selisih jaraknya ditambahkan ke total jarak pulau yang bersangkutan
 - Jika tidak di satu pulau
 - ◆ selisih jaraknya ditambahkan ke total jarak kapal ferry
 - Nama *vertex* sekarang dicetak



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 0

Total jarak di pulau W2: 0

Total jarak di pulau W3: 0

Total jarak kapal ferry: 0

Malia

100

100

Korkyra

0

106.39

27

106.8

29

Knossos

109.96

19



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 2.23

Total jarak di pulau W2: 0

Total jarak di pulau W3: 0

Total jarak kapal ferry: 0

Malia

100

100

Korkyra

0

106.39

27

106.8

29

Knossos

109.96

19



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 7.23

Total jarak di pulau W2: 0

Total jarak di pulau W3: 0

Total jarak kapal ferry: 0

Malia

100

100

Korkyra

0

106.39

27

106.8

29

Knossos

109.96

19



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 9.46

Total jarak di pulau W2: 0

Total jarak di pulau W3: 0

Total jarak kapal ferry: 0

Malia

100

100

Korkyra

0

106.39

27

106.8

29

Knossos

109.96

19



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 9.46

Total jarak di pulau W2: 0

Total jarak di pulau W3: 0

Total jarak kapal ferry: 100

Malia

100

100

Korkyra

0

106.39

27

106.8

29

Knossos

109.96

19



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 9.46

Total jarak di pulau W2: 9.22

Total jarak di pulau W3: 0

Total jarak kapal ferry: 100

Malia

100

100

Korkyra

0

106.39

27

106.8

29

109.96

19

119.18

Knossos



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 9.46

Total jarak di pulau W2: 12.79

Total jarak di pulau W3: 0

Total jarak kapal ferry: 100

Malia

100

100

Korkyra

0

106.39

106.8

29

27

109.96

19

Knossos

119.18



Lindos

228.65

226.42

1

221.42

219.18

Kamejros

Total jarak di pulau W1: 9.46

Total jarak di pulau W2: 13.2

Total jarak di pulau W3: 0

Total jarak kapal ferry: 100

119.18

Knossos

100

10 June 2017

Tugas Akhir - KI1502

123

Malia

100

100

Korkyra

0

106.39

27

106.8

29

109.96

19



Lindos

228.65

226.42

1

10

221.42

219.18

Kamejros

Total jarak di pulau W1: 9.46

Total jarak di pulau W2: 19.59

Total jarak di pulau W3: 0

Total jarak kapal ferry: 100

Malia

100

100

Korkyra

0

106.39

27

106.8

29

Knossos

109.96

19



Lindos

228.65

226.42

221.42

219.18

Kamejros

Total jarak di pulau W1: 9.46

Total jarak di pulau W2: 19.59

Total jarak di pulau W3: 0

Total jarak kapal ferry: 200

Malia

100

100

0

106.39

27

106.8

29

Knossos

119.18

109.96

19



Lindos

228.65

226.42

1

10

221.42

219.18

Kamejros

Total jarak di pulau W1: 10

Total jarak di pulau W2: 20

Total jarak di pulau W3: 0

Total jarak kapal ferry: 200

Malia

100

100

Korkyra

0

106.39

27

106.8

29

Knossos

109.96

19



Total jarak di pulau W1: 230

Lindos

228.65

226.42

1

221.42

219.18

Kamejros

119.18

Knossos

10 June 2017

127

19

29

27

0

100

100

106.39

106.8

Malia

Tugas Akhir - KI1502



ITS

Institut
Teknologi
Sepuluh Nopember

PENDAHULUAN

TINJAUAN PUSTAKA

STRATEGI PENYELESAIAN

UJI COBA

KESIMPULAN



ITS

Institut
Teknologi
Sepuluh Nopember

UJI COBA KEBENARAN

17025692	2016-06-01 05:32:52	The Archipelago	accepted edit ideone.it	0.26	10M	C++ 5
----------	---------------------	-----------------	----------------------------	------	-----	-------

- Implementasi yang telah dilakukan mendapat umpan balik **Accepted** dari situs penilaian SPOJ.

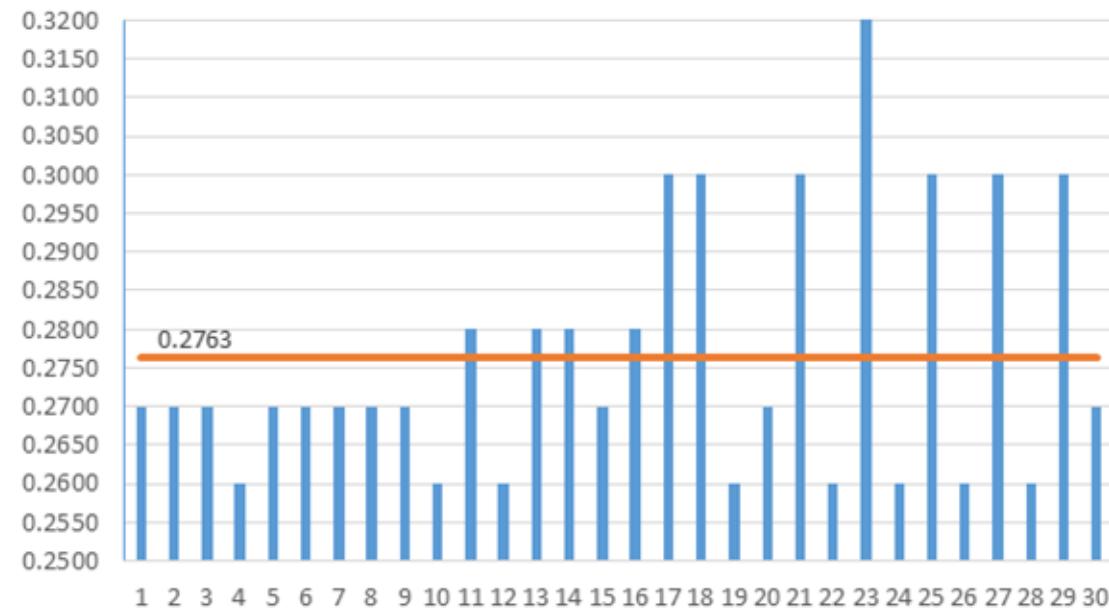
RANK	DATE	USER	RESULT	TIME	MEM	LANG
1	2016-04-10 09:54:17	Reva Yoga Pradana	accepted	0.26	11M	C++ 5

- Hasil implementasi mendapat peringkat pertama (dari segi *running time*) pada situs SPOJ The Archipelago.



UJI COBA EFISIENSI

Uji Coba pada Situs SPOJ Sebanyak 30 Kali



- Waktu yang dibutuhkan program minimum 0,26 detik, maksimum 0,32 detik dan rata-rata 0,2763 detik. Memori yang dibutuhkan program konstan sebesar 10 MB.



ITS

Institut
Teknologi
Sepuluh Nopember

PENDAHULUAN

TINJAUAN PUSTAKA

STRATEGI PENYELESAIAN

UJI COBA

KESIMPULAN



KESIMPULAN

- Implementasi algoritma Dijkstra dan Visibility Graph Naive dapat menyelesaikan permasalahan The Archipelago dengan benar.
- Kompleksitas sebesar $O(|E| + V \log V)$ pada Algoritma Dijkstra, dan kompleksitas sebesar $O(N^3)$ pada Visibility Graph Naive cukup untuk menyelesaikan permasalahan The Archipelago.
- Optimasi-optimasi yang telah dilakukan dapat membantu mengurangi running time dari metode visibility graph naive yang sebenarnya memiliki kompleksitas cukup besar.
- Waktu yang dibutuhkan program minimum 0,26 detik, maksimum 0,32 detik dan rata-rata 0,2763 detik. Memori yang dibutuhkan program rata-rata sebesar 10.533 MB.



ITS

Institut
Teknologi
Sepuluh Nopember

TERIMA KASIH



ITS

Institut
Teknologi
Sepuluh Nopember

DAFTAR PUSTAKA

NO

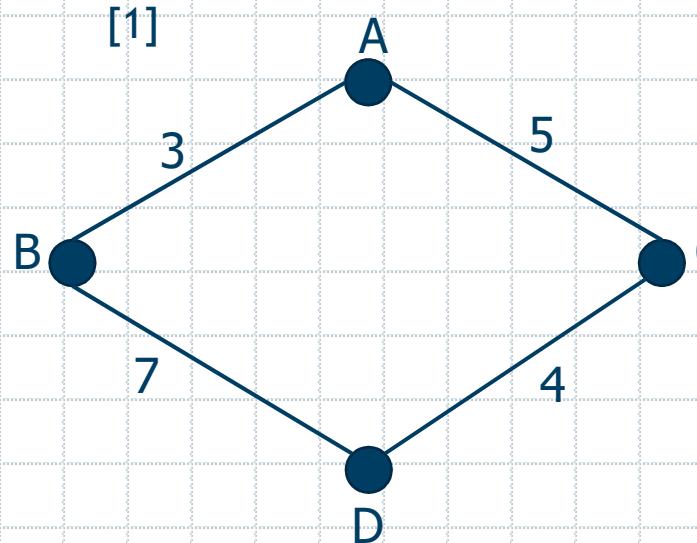
REFERENSI

- [1] A. Levitin, *The Design & Analysis of Algorithms*, 3rd ed., Pearson Education, Inc., 2012.
- [2] M. de Berg, O. Cheong, M. van Kreveld and M. Overmars, *Computational Geometry*, 3rd ed., Berlin Heidelberg: Springer-Verlag, 2008.
- [3] S. Halim and F. Halim, *Competitive Programming 2*, Singapore: Lulu Publisher, 2011.
- [4] cplusplus.com, "priority_queue - C++ Reference," [Online]. Available: http://wwwcplusplus.com/reference/queue/priority_queue/. [Accessed 7 June 2016].
- [5] cplusplus.com, "greater - C++ Reference," [Online]. Available: <http://wwwcplusplus.com/reference/functional/greater/>. [Accessed 7 June 2016].
- [6] L. Kuszner, "SPOJ.com - Problem ARCHPLG," 15 March 2006. [Online]. Available: <http://www.spoj.com/problems/ARCHPLG/>.

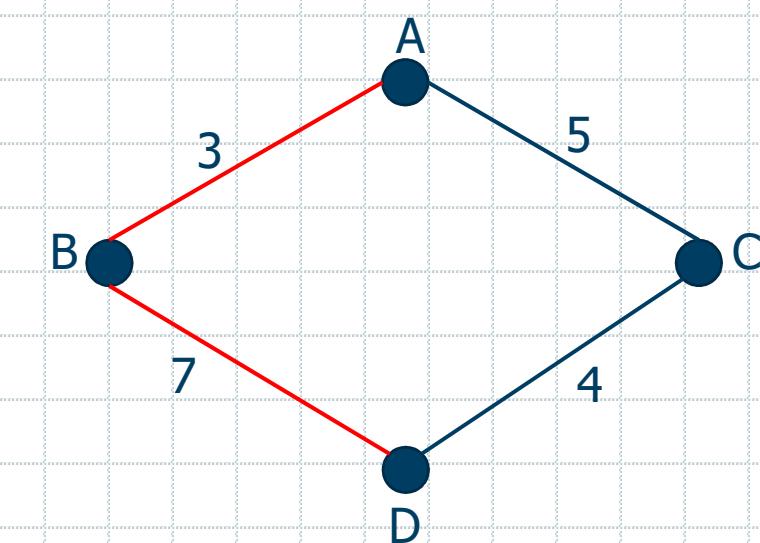


DEFINISI UMUM

- *Weighted Undirected Graph* adalah graf dimana *edge* yang terdapat didalamnya memiliki bobot namun tidak memiliki arah. [1]
- Rute terpendek yang dimaksud pada *Weighted Undirected Graph* adalah rute dengan jumlahan bobot *edge* paling minimum dari suatu *node* menuju *node* lain.



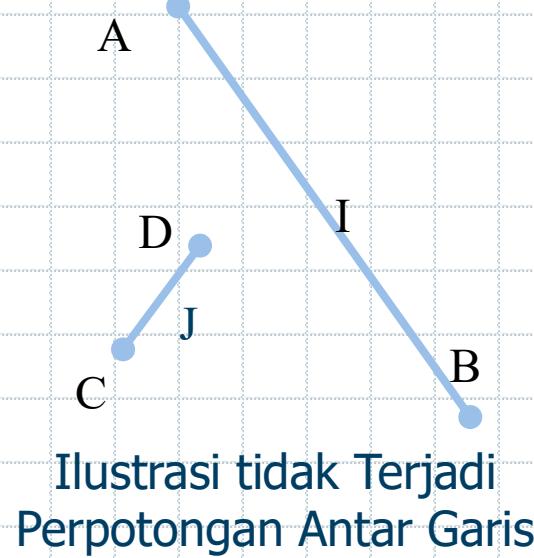
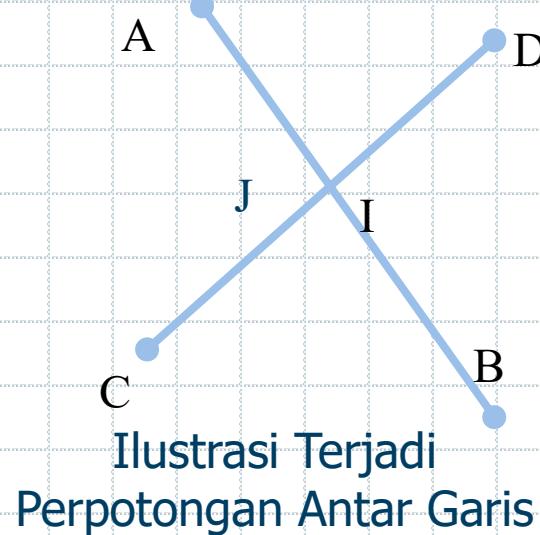
Weighted Undirected Graph



Rute terpendek dari *node* A menuju *node* D

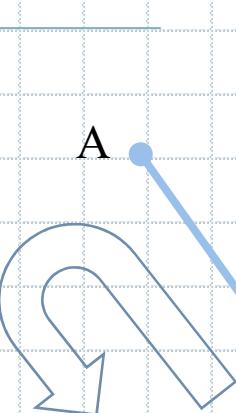


CHECK INTERSECTION BETWEEN SEGMENT LINES





CHECK INTERSECTION BETWEEN SEGMENT LINES



I

A

C

B

A

D

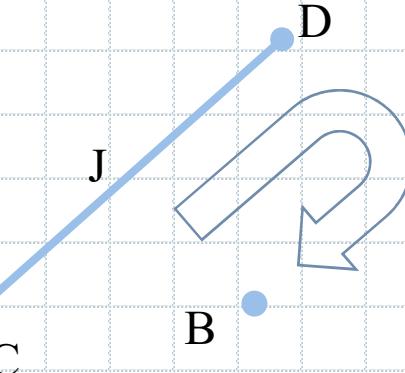
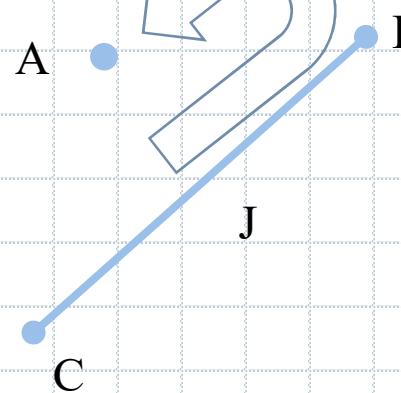
I

B

Syarat 1:
Arah dari garis I ke titik C berkebalikan dengan arah ke titik D



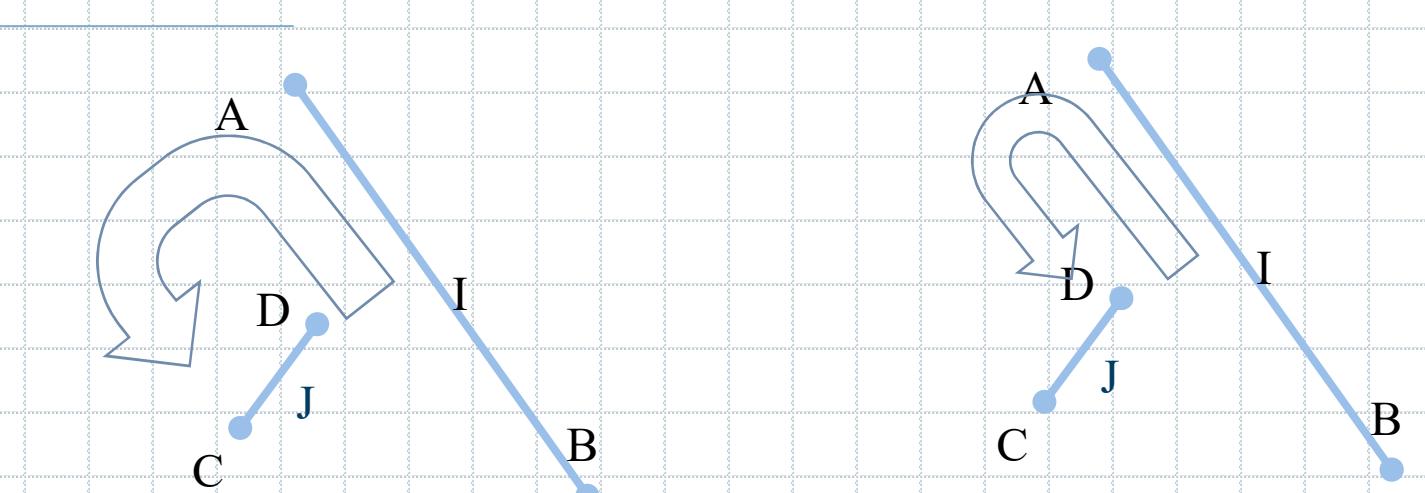
CHECK INTERSECTION BETWEEN SEGMENT LINES



Syarat 2:
Arah dari garis J ke titik A berkebalikan dengan arah ke titik B



CHECK INTERSECTION BETWEEN SEGMENT LINES



Syarat 1:

Arah dari garis I ke titik C tidak berkebalikan dengan arah ke titik D \rightarrow tidak terjadi perpotongan



ITS

Institut
Teknologi
Sepuluh Nopember

CHECK INTERSECTION BETWEEN SEGMENT LINES

- Untuk pengecekan arah perputarannya, digunakan perhitungan 4 kali cross product seperti berikut.

- int o1 = orientation(p1, q1, p2);
- int o2 = orientation(p1, q1, q2);
- int o3 = orientation(p2, q2, p1);
- int o4 = orientation(p2, q2, q1);
- p1 = line1.x, q1 = line1.y, p2 = line2.x, q2 = line2.y
- Jika $o1 / o2 / o3 / o4 == 0 \rightarrow$ collinear \rightarrow tidak ada intersection
- Jika $o1 != o2 \rightarrow$ syarat 1 terpenuhi
- Jika $o3 != o4 \rightarrow$ syarat 2 terpenuhi



ITS

Institut
Teknologi
Sepuluh Nopember

CHECK INTERSECTION BETWEEN SEGMENT LINES

- function orientation(point p, point q, point r)

{

```
    return (q.y - p.x) * (r.x - q.x) - (q.x - p.x) * (r.y - q.y);
```

}



Hasil Pengujian Sebelum Optimasi

ID	DATE	PROBLEM	RESULT	TIME	MEM	LANG
16264128	2016-02-11 01:40:03	The Archipelago	accepted	4.08	14M	C++14
16262449	2016-02-10 18:21:37	The Archipelago	accepted	3.88	14M	C++14
16262190	2016-02-10 17:37:30	The Archipelago	accepted	3.94	14M	C++14
16262083	2016-02-10 17:20:36	The Archipelago	time limit exceeded	-	14M	C++14
16261796	2016-02-10 16:22:44	The Archipelago	time limit exceeded	-	14M	C++14
16261774	2016-02-10 16:17:58	The Archipelago	time limit exceeded	-	14M	C++14
16261689	2016-02-10 16:04:29	The Archipelago	time limit exceeded	-	7.0M	C++14
16261656	2016-02-10 15:57:49	The Archipelago	time limit exceeded	-	7.9M	C++14
16261639	2016-02-10 15:54:53	The Archipelago	time limit exceeded	-	6.6M	C++14
16260795	2016-02-10 13:17:50	The Archipelago	time limit exceeded	-	6.7M	C++14
16259955	2016-02-10 10:25:18	The Archipelago	wrong answer	3.96	14M	C++14
16259933	2016-02-10 10:19:53	The Archipelago	time limit exceeded	-	14M	C++14
16259926	2016-02-10	The Archipelago	time limit exceeded	-	14M	C++14



Hasil Pengujian Sebelum Optimasi

16252461	2016-02-09 01:19:33	The Archipelago	time limit exceeded	-	14M	C++14
16247665	2016-02-08 03:31:38	The Archipelago	wrong answer	3.93	14M	C++14
16246397	2016-02-07 19:09:51	The Archipelago	wrong answer	0.81	14M	C++14
16244905	2016-02-07 14:32:40	The Archipelago	time limit exceeded	-	14M	C++14
16244820	2016-02-07 14:14:47	The Archipelago	time limit exceeded	-	14M	C++14
16244737	2016-02-07 14:05:30	The Archipelago	time limit exceeded	-	14M	C++14
16244655	2016-02-07 13:58:05	The Archipelago	time limit exceeded	-	14M	C++14



ALGORITMA DIJKSTRA

- Salah satu algoritma yang dapat menyelesaikan permasalahan rute terpendek dengan kompleksitas waktu sebesar $O(|E| + V \log V)$.
- Algoritma Dijkstra menentukan rute terpendek dari *vertex* awal atau sumber menuju ke seluruh *vertex* pada graf.
- Berikut adalah langkah-langkah dari algoritma Dijkstra:
 1. Tetapkan nilai jarak awal untuk tiap *vertex* dari *vertex* sumber dengan nilai sebesar ∞ , kecuali untuk *vertex* sumber, nilainya sebesar 0.
 2. Tandai tiap *vertex* bahwa “belum pernah dikunjungi”, kecuali *vertex* sumber, ditandai dengan “sudah dikunjungi”.
 3. Terdapat *priority queue* yang menyimpan nilai jarak untuk tiap *vertex* yang telah dicek / dikunjungi, dari *vertex* sumber.



ALGORITMA DIJKSTRA

4. Untuk langkah awal, masukan *vertex sumber* pada *priority queue* dengan nilai jarak sebesar 0.
5. Kemudian, dilakukan iterasi untuk tiap isi dari *priority queue*.
6. Tiap iterasi akan memanggil elemen teratas, yaitu *vertex* dengan total jarak terkecil dari *vertex sumber*.
 - Iterasi akan berhenti apabila:
 - *priority queue* kosong
 - elemen teratas = *vertex tujuan*
7. Hitung nilai jarak untuk tiap tetangga dari *vertex* sekarang (elemen teratas tadi).
 - ◆ jarak = bobot *edge* + jarak pada *vertex* sekarang.



ITS

Institut
Teknologi
Sepuluh Nopember

ALGORITMA DIJKSTRA

7. Jika:

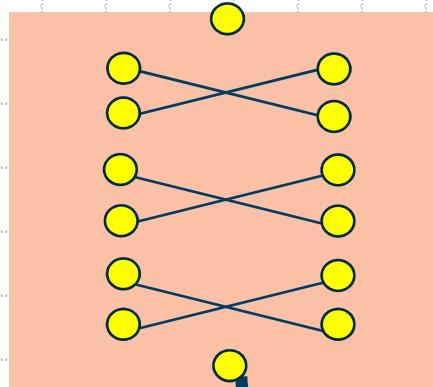
- ◆ jumlahan tersebut < nilai jarak pada *vertex* tetangga
 - Nilai jarak pada *vertex* tetangga = jumlahan tersebut
 - *Vertex* tetangga masuk ke *priority queue*

8. Tandai *vertex* sekarang dengan “sudah dikunjungi”.

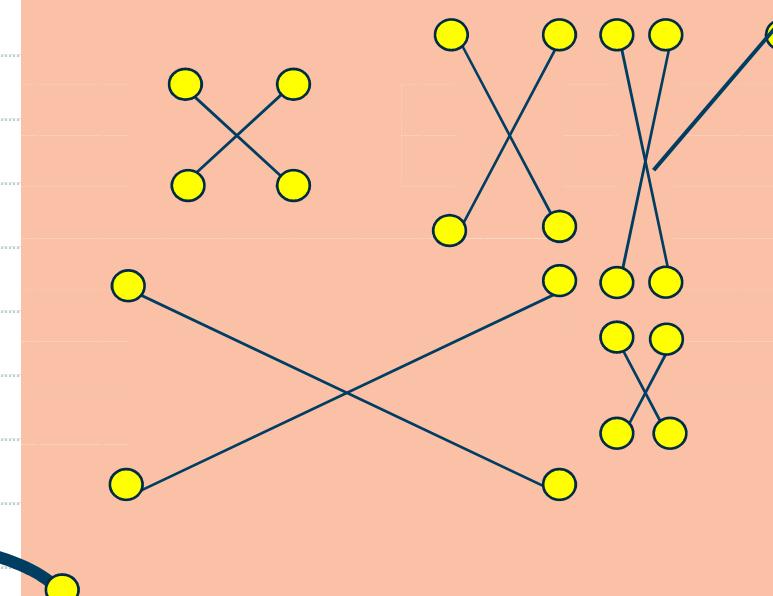
9. Saat iterasi tersebut berhenti, maka nilai jarak pada *vertex* tujuan adalah total jarak dari rute terpendek pada permasalahan tersebut.



VISIBILITY GRAPH NAIVE



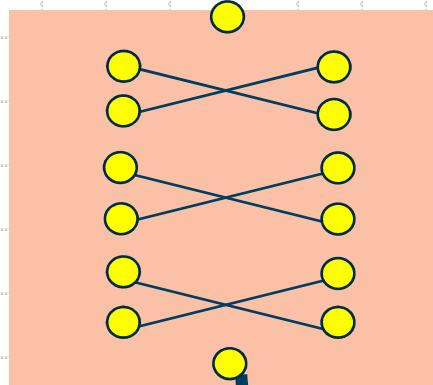
100



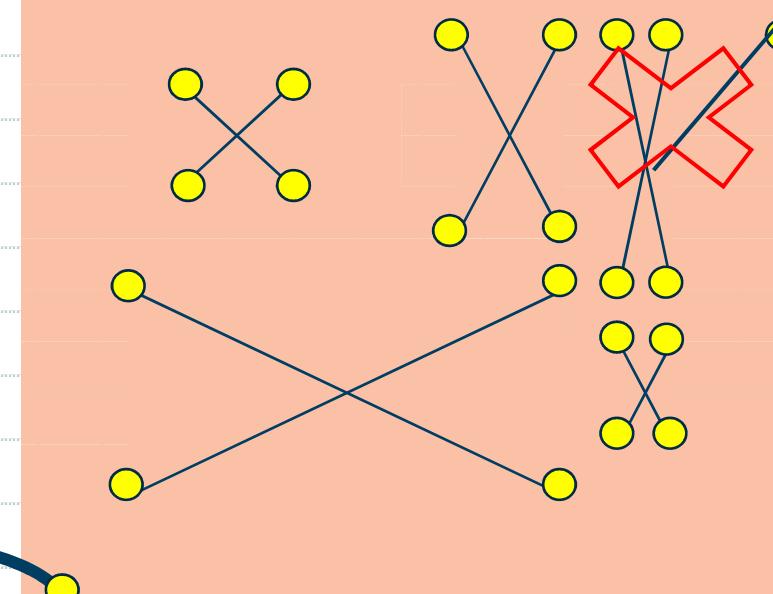
100



VISIBILITY GRAPH NAIVE



100



100



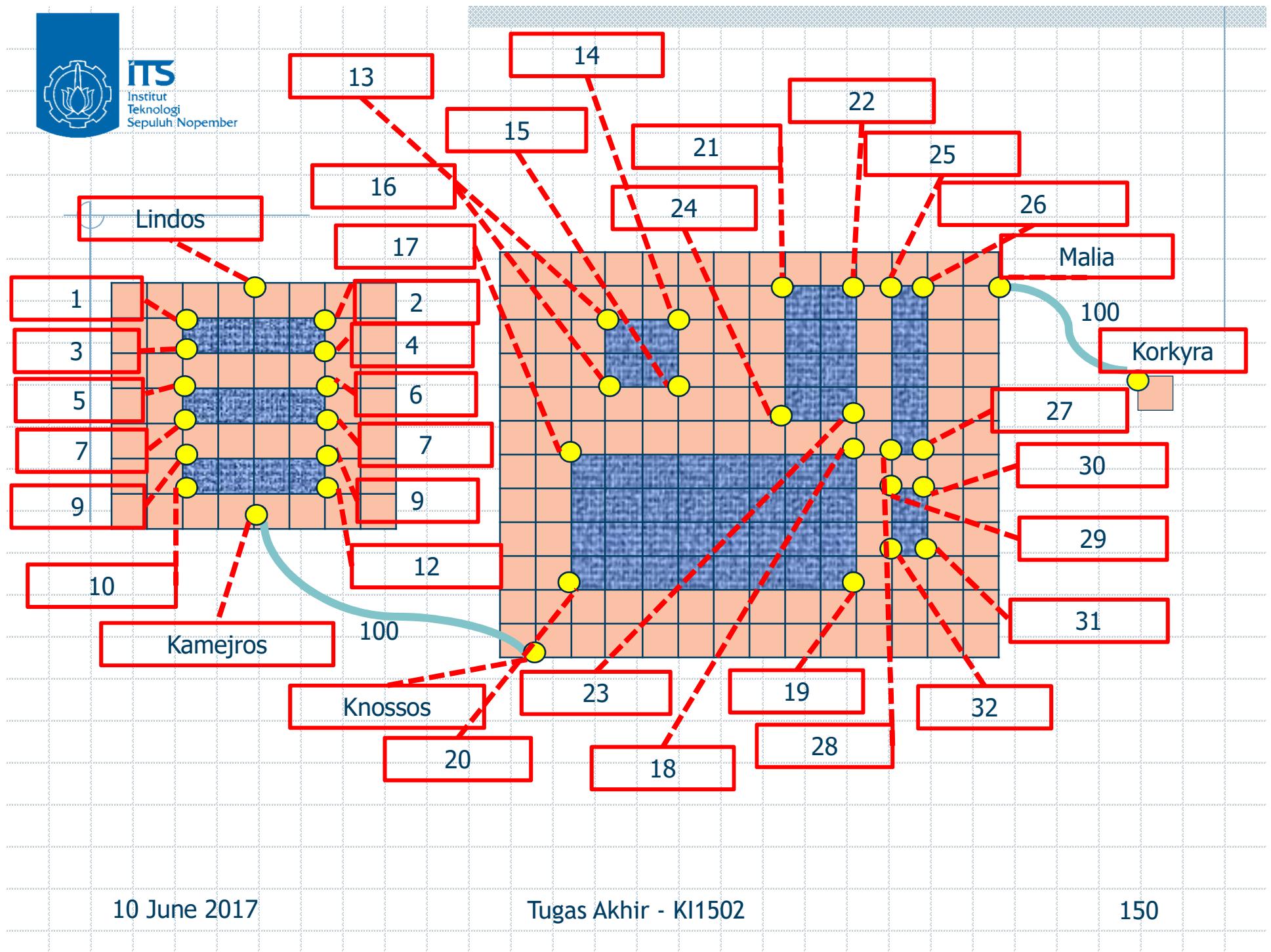
ITS

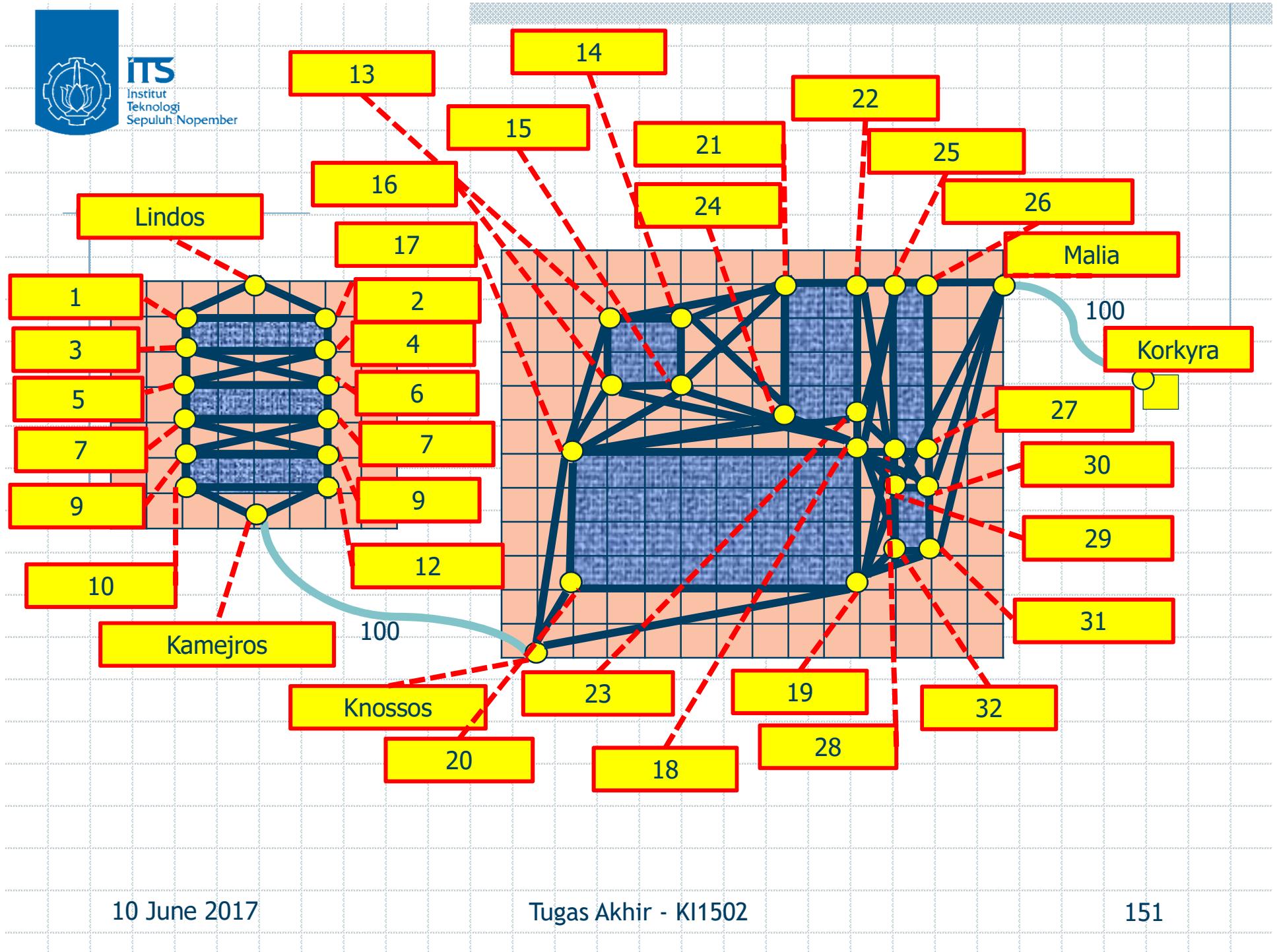
Institut
Teknologi
Sepuluh Nopember

STRATEGI PENYELESAIAN

1. Penggunaan Visibility Graph Naive

- Langkah paling mudah untuk menyelesaikan permasalahan ini adalah dengan mengecek visibilitas dari semua vertex, di semua pulau. Kemudian, dicari rute terpendeknya.







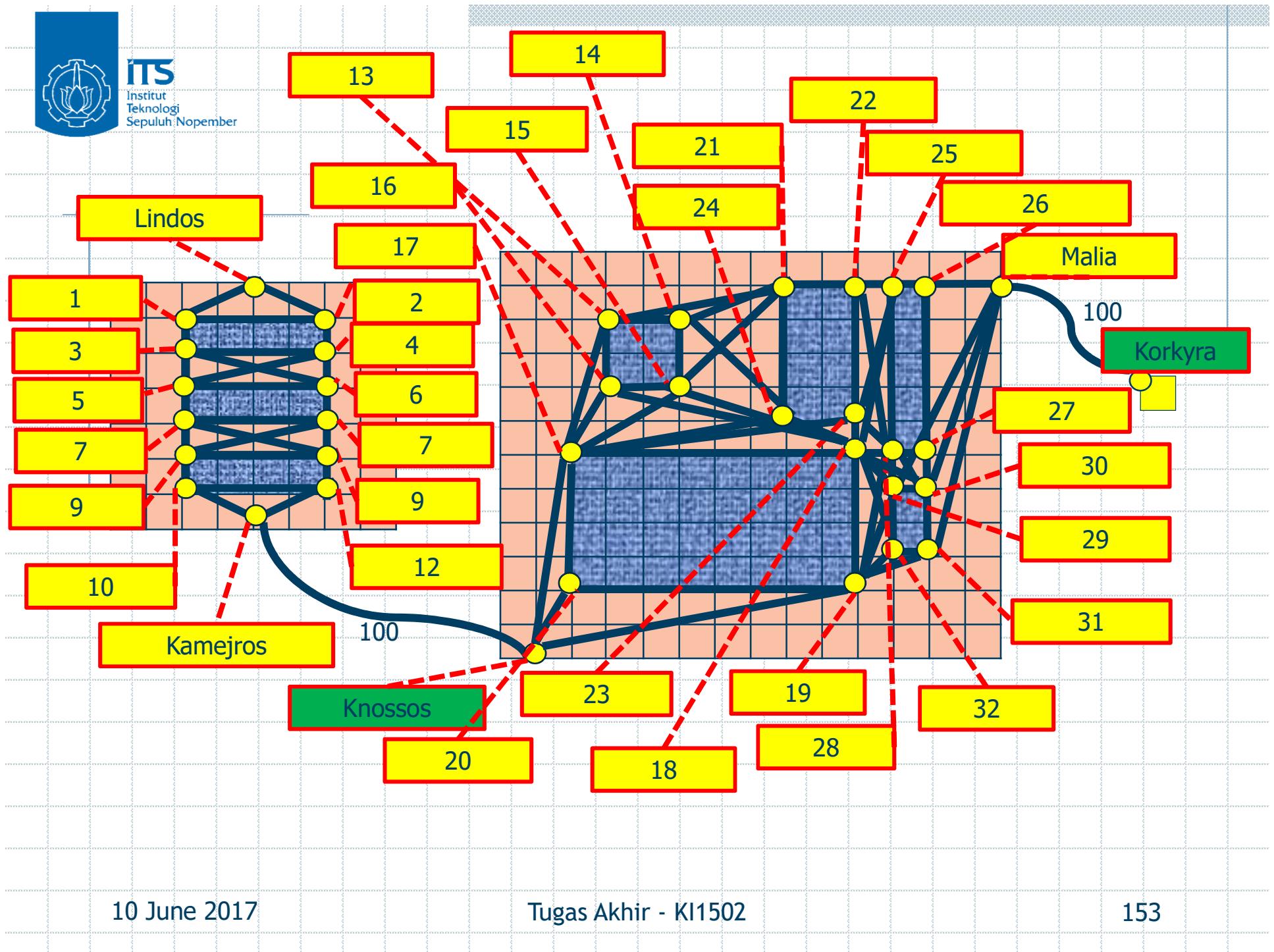
ITS

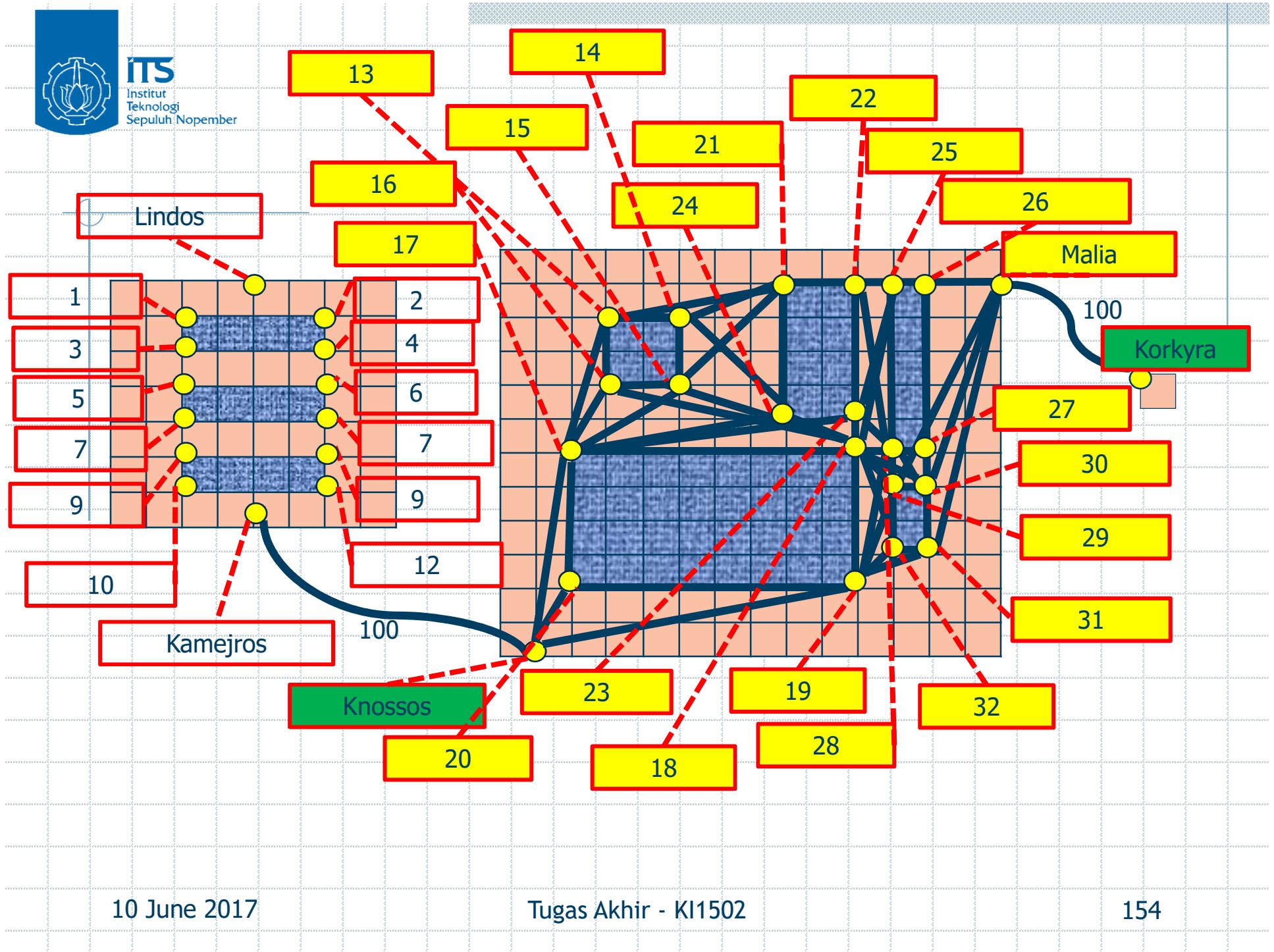
Institut
Teknologi
Sepuluh Nopember

STRATEGI PENYELESAIAN

1. Penggunaan Visibility Graph Naive

- Namun, hal tersebut kurang efektif.
- Misal yang dicari adalah rute terpendek dari Korkyra W3 ke Knossos W2







ITS

Institut
Teknologi
Sepuluh Nopember

Euclidean Distance

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



SARAN

- Saran yang diberikan adalah selanjutnya dapat digunakan metode *visibility graph* yang lebih cepat / lebih baik dari metode *naive*.