**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY



**Diskrétna simulácia**

Semestrálna práca S1 - Robot

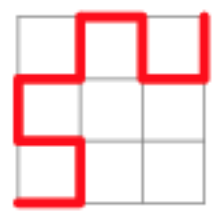
Dávid Pavličko

5ZYS12

2020/2021

# **Zadanie práce**

Študenti sa rozhodli prihlásiť na súťaž robotov. Robot sa pohybuje po grafe, ktorý je tvorený štvorcovou sieťou. Na obrázku vidíme sieť s rozmermi 4x4 vrcholov (24 hrán), pričom robot vyštartoval z pozície na súradniciach [0, 0]. Cieľom je, aby robot prešiel čo najdlhšiu vzdialenosť, pričom každý vrchol grafu môže navštíviť najviac jeden krát. Pohyb robota končí, keď navštívi nejaký vrchol opakovane.



Študenti si uvedomujú, že existujú postupy, ktoré umožnia robotovi efektívne prejsť čo najväčší počet hrán. Keďže súťaž je už za niekoľko dní nie je už čas robota zdokonaliť. Ich robot sa po prejdení hrany pohne vždy náhodne jedným z dostupných smerov (napr. doľava, dopredu, doprava ak nie je pri okraji).

Vytvorte počítačový model uvedenej situácie. Rozmery obdĺžnikového priestoru sú vstupnými parametrami modelu (počet vrcholov na x-ovej a y-ovej súradnici), rovnako tak ako štartovacia súradnica pohybu. Prvý pohyb robí robot náhodne do jedného z dostupných smerov. Napr. ak štartuje zo súradnice [0, 0] vyberá si z dvoch ciest.

S využitím metódy Monte Carlo vykonajte s týmto modelom experimenty tak, aby ste boli schopní zodpovedne vypracovať nasledovné úlohy:

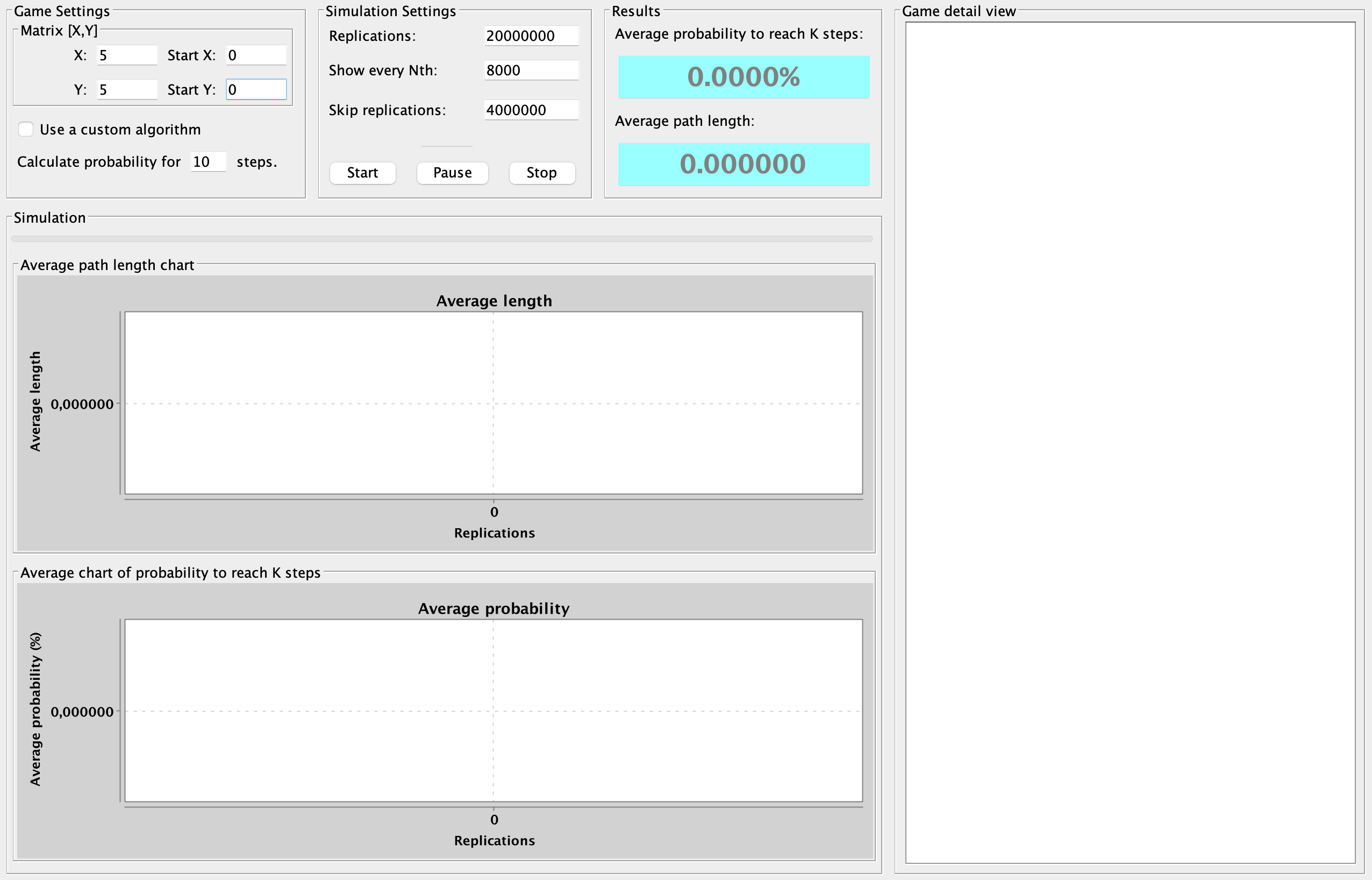
1. A.)  Koľko krokov priemerne robot urobí, kým sa zastaví? Graficky (na grafe v programe) dokumentujte “vývoj” ustaľovania tejto hodnoty s rastúcim počtom uskutočnených replikácií.
2. B.)  Aká je pravdepodobnosť, že urobí viac ako ***K*** krokov (***K*** je zadaný parameter). Graficky (na grafe v programe) dokumentujte “vývoj” ustaľovania tejto pravdepodobnosti s rastúcim počtom uskutočnených replikácií.

C.)Upravte Váš model o**Vami navrhnutú** stratégiu, ktorá umožní robotovi prejsť čo najväčšiu vzdialenosť

Zabezpečte (napr. pomocou vhodných nastavení), aby grafy mali čo najväčšiu čitateľnosť zobrazovaných dát a mali aj reálny prínos pre užívateľa. Simulácia sa musí dať predčasne zastaviť a zobraziť dosiahnuté výsledky. Pozastavenie nie je potrebné implementovať.

Implementujte všeobecné jadro pre statické modelovanie metódou Monte Carlo. Pri implementácií semestrálnej práce dodržte oddelenie užívateľského prostredia od jadra aplikácie. V semestrálnej práci je na generovanie čísel dovolené používať iba v danom jazyku štandardné knižnice (napr. triedu Random v jazyku java a C#).

# **Ovladanie aplikácie (GUI)**



**Herné nastavenia:**

* Nastavenie veľkosti hernej plochy X\*Y a začiatočnej pozície robota v nej.
* Možnosť voľby použitia vlastného algoritmu prechádzania od vrcholu k vrcholu robotom.
* Zadávanie konštanty K, z ktorej vypočítavam pravdepodobnosť v percentách dosiahnutia daného počtu prejdených vrcholov v hre.

**Simulačné nastavenia:**

* Počet replikácií v jednej hre.
* Zobrazenie každej N-tej replikácie do grafov.
* Preskočenie N replikácií od začiatku, pre ustálenie hodnôt.

**Tlačidlá:**

* Štart, pauza, zastavenie

**Výsledky:**

* Priemerná pravdepodobnosť prekročenia dĺžky K dosiahnutých vrcholov v hre, zobrazená v percentách.
* Priemerná robotova dĺžka cesty

**Grafy:**

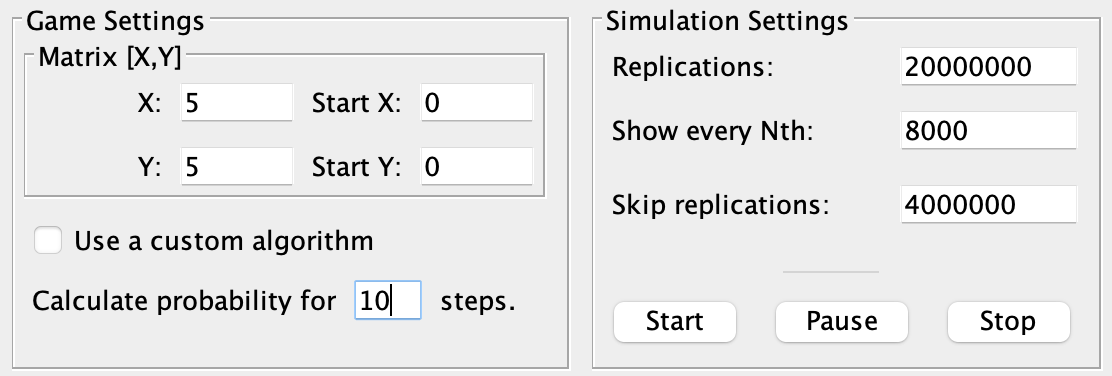
* Graf ustaľovania dĺžky dráhy, ktorú robot v inicializačných nastaveniach prešiel.
* Graf ustaľovania pravdepodobnosti prejdenia väčšej dĺžky ako K

# **Úlohy**

**1. Koľko krokov priemerne robot urobí, kým sa zastaví?**

Robot sa môže pohybovať maximálne v štyroch smeroch. Ak robot narazí na vrchol, ktorý už v ceste navštívil, hra končí a tento vrchol sa už nezapočítava. Ak robot môže cestovať len jedným smerom, tento smer použije. V prípade, že je robot v pozícií, že má možnosť výberu z dvoch, troch alebo štyroch smeroch v prípade štartu, pre každý smer je použitá vlastná inštancia triedy Random, aby sa zabezpečilo náhodné rozdelenie pravdepodobností.

**Inicializačný stav hry:**



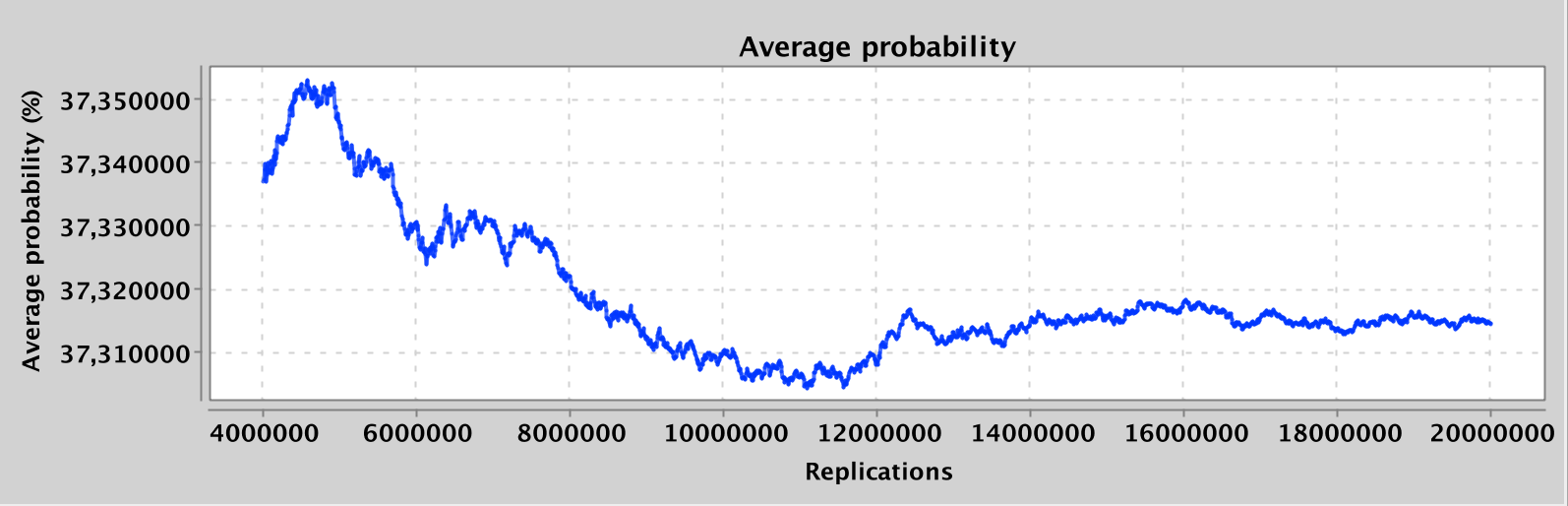
**Výsledky:**

Graf ustaľovania dĺžky trasy:



**2. Aká je pravdepodobnosť, že urobí viac ako K krokov?**

Graf ustaľovania pravdepodobnosti prekročenia K(v našom prípade 10) krokov:



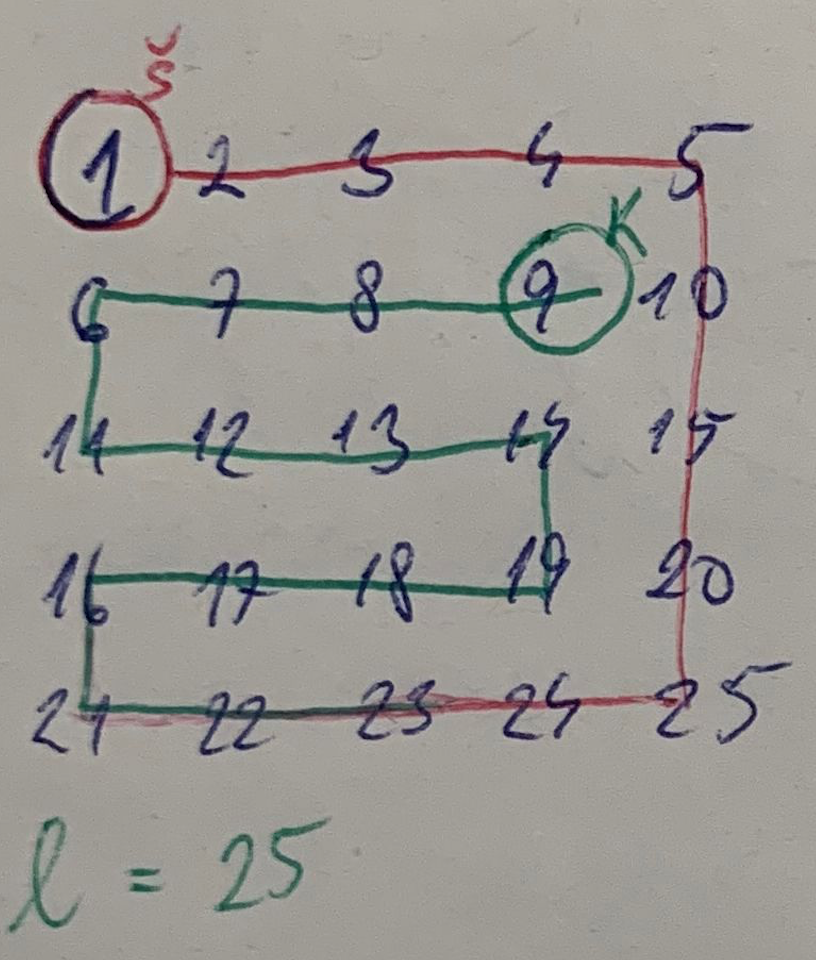
**3. Vlastný algoritmus určenia smeru:**

Za vlastný spôsob som použil upravenú metódu pohybu domáceho robotického vysávača. Keďže robot v hre nemôže prejsť dvakrát rovnakým vrcholom a nevie ani veľkosť plochy, alebo inak si ju vyrátať vzhľadom na pozíciu, pretože náš robot je jednoduchý a vidí len do vzdialenosti 1 vrchol okolo seba, som sa rozhodol určovať smer pohybu v pevne danom poradí.

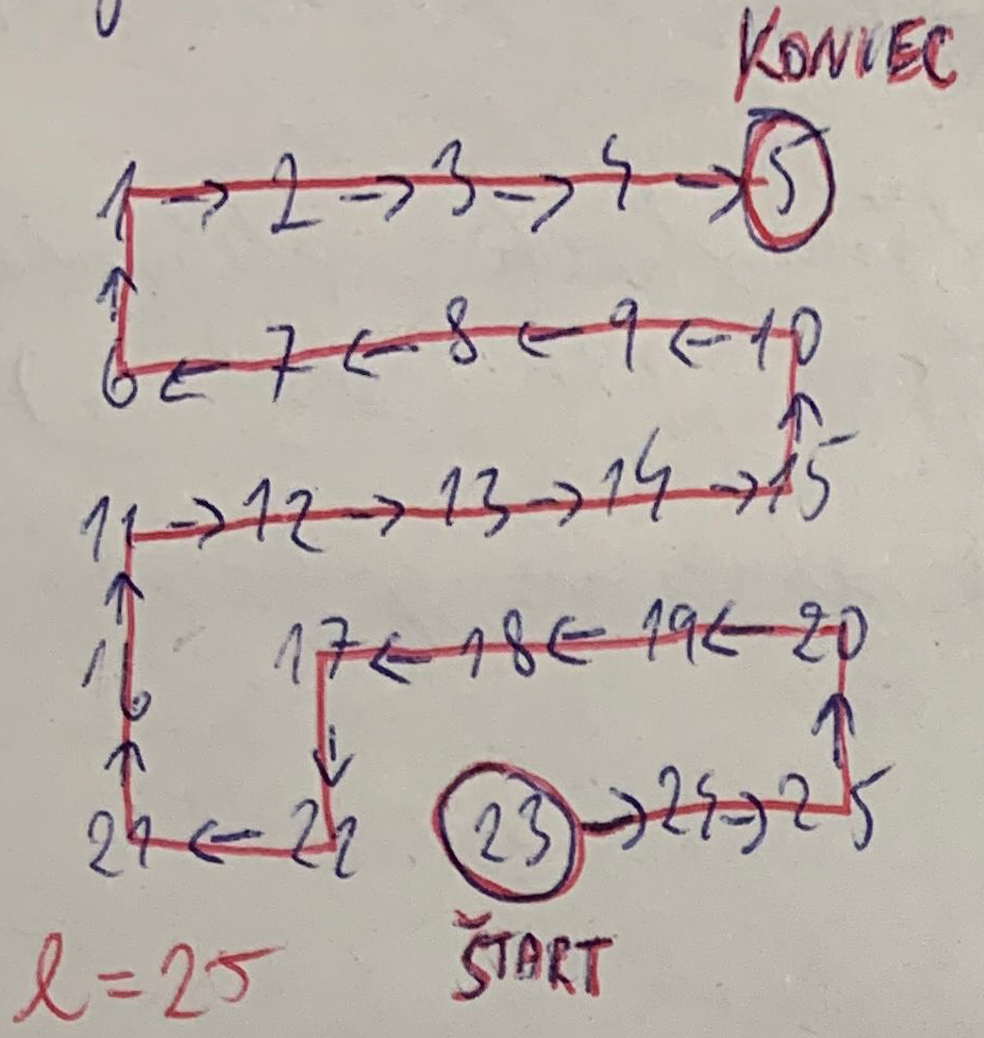
Najskôr sa pokúsi prejsť čo najväčší poloblúk, teda ide vpravo až kým nenarazí na okraj plochy a potom dole. Ak už nemôže ísť ani vpravo ani dole, tretia voľba je ísť vľavo, ak ani tam, tak zvolí smer nahor. Ak nie je možné ísť ani jedným smerom, hra skončila.

Táto taktika veľmi závisí od štartovnej pozície a preto si ukážeme niektoré prípady, začneme rovnakým nastavením ako v predošlých úlohách.

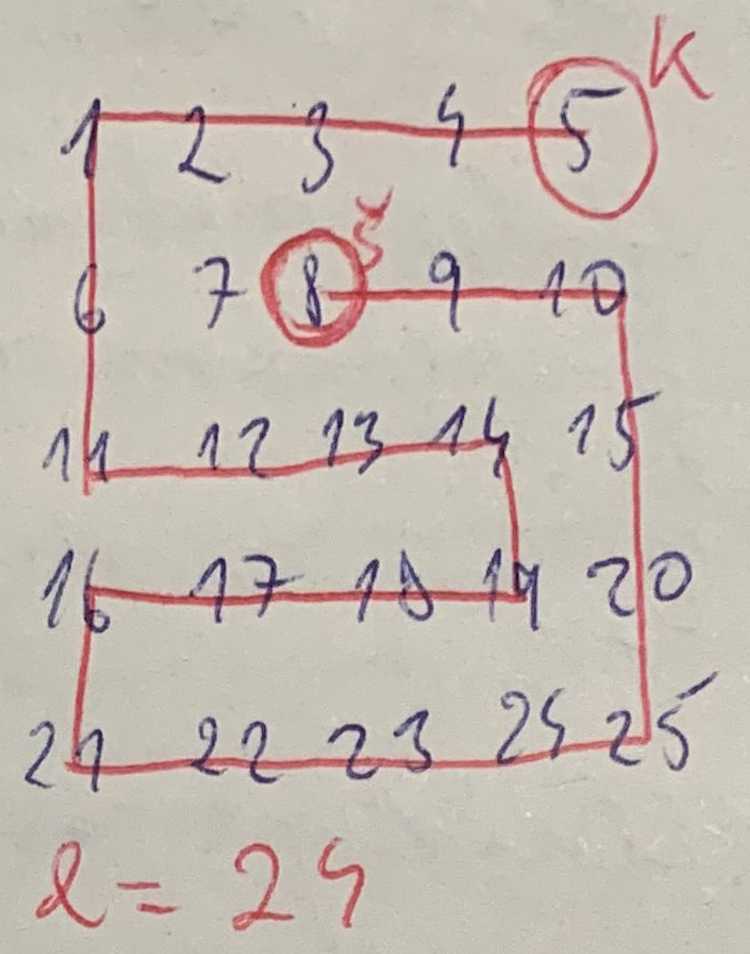
* [0,0] => 1



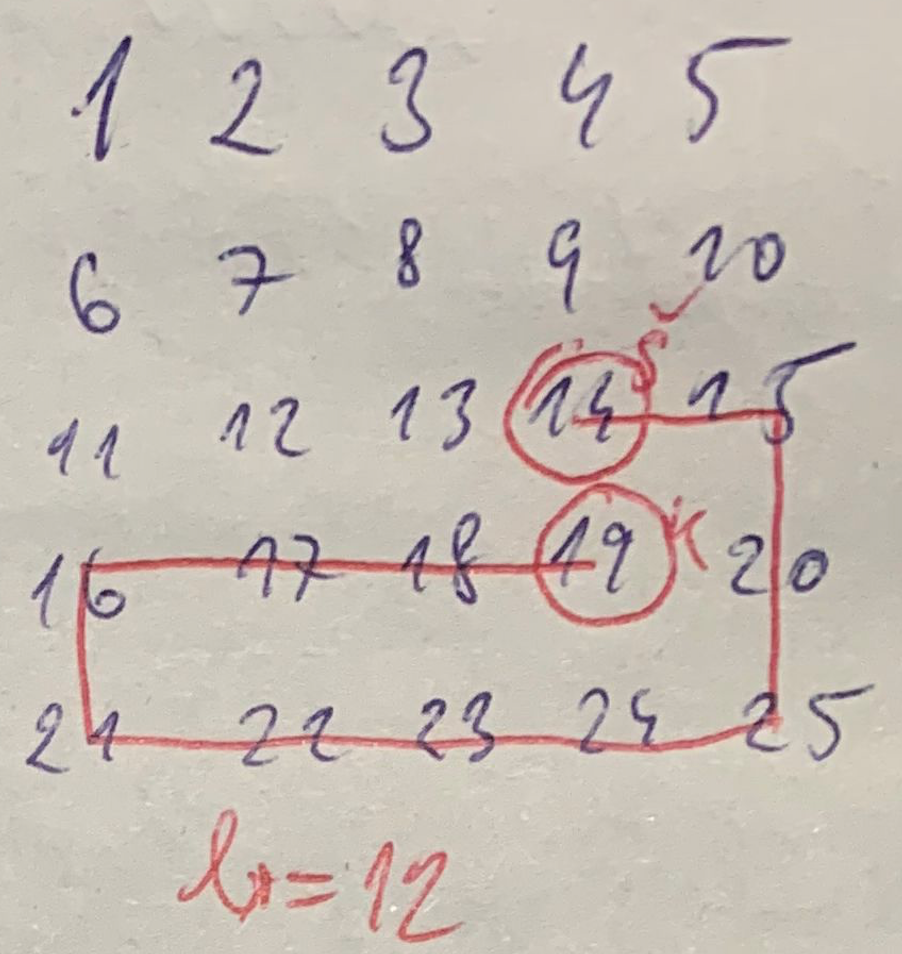
* [4,2] => 23



* [1,2] => 8



* [2,3] => 14



# **Použité knižnice**

Pre vykresľovanie grafov v reálnom čase som použil knižnicu XChart, ktorú je možné nájsť tu: <https://github.com/knowm/XChart>