

平成 18 年度

東京大学情報理工学系研究科創造情報学専攻

プログラミング

注意事項

1. 試験の合図まで，この問題冊子を開いてはいけない。
2. この表紙の下にある受験番号欄に受験番号を記入しなさい。
3. 解答用紙，下書き用紙がそれぞれ 1 枚配られる。解答用紙に受験番号を記入しなさい。
4. やや長文の問題であるが，解を誘導するように書かれているので，一度通読してから解答するとよい。
5. 各自の PC を用いて指定されたプログラムを作成せよ。ただし，最初の 2 問は解答用紙に解答すること。そのあとの問題も（手書きの図を書くなど）必要であれば，解答用紙に解答の一部を書いてもよい。
6. プログラミング言語は各自の得意なものを使用しなさい。
7. プログラミング言語のマニュアルは 1 冊に限り実技中に参照してもよい。
8. 試験終了時に各受験者に USB メモリが渡される。受験番号名のディレクトリ/フォルダを作成の上，作成したプログラムおよび関連ファイルをその下にコピーし，さらにこのディレクトリ/フォルダを USB メモリにコピーしなさい。
9. コピー終了後，USB メモリ，解答用紙，下書き用紙を回収する。問題冊子を持ち帰ってはならない。
10. 回収後，試験監督が周回し，各受験者の結果を簡単に確認する。そのまま座席で待機しなさい。全員の確認が終わるまで部屋を出てはいけない。
11. 口述試験中にプログラムの動作をより精密に確認する。試験監督の簡易確認のあと，各自の PC 上でプログラムがなるべくすぐに実行できるようにしておきなさい。
12. 全員の簡易確認が終了した後，各自の PC とこの問題冊子を指示された部屋へ運び，その部屋から退出しなさい。

受験番号 _____

Sprouts は 1967 年 2 月 21 日, Life ゲームの創造で有名な John H. Conway が大学院生の Michael S. Paterson と, 紙と鉛筆で遊べる新しいゲームを生み出そうというお茶飲み話をしながら創造したゲームである.

n 点 Sprouts のルールは以下の通り. 最初に n 個の点 (spot) を紙の上に書く. 先手, 後手は交互に手を打つ. 最初に手を打てなくなったほうが負けである.

Sprouts の手とは, 1 つの点をもう 1 つの点 (自分自身でもよい) と線 (line) で結び, その線の上に新たに点を 1 つ付け加えることである. ただし,

- (1) 線はどのような形でもよいが, 自分自身と交わったり, すでに書かれている線と交わったり, 他の既存の点と交わってはいけない.
- (2) どの点も, それにつながる線は 3 本を超えてはならない. なお, 新たに付け加えられた点はその時点で 2 本の線につながっている.

次ページの図 1 に, 3 点 Sprouts において先手が図 1(8) で勝利した進行例を示す. 最初からあった 3 点には 1, 2, 3 の番号がついている. 各競技者が点を付け加えるたびに, それに続く番号を順に振る. 領域が分割されたときは, どちらか一方に順次新しい番号を振る.

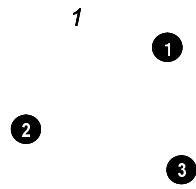
問 1 1 点 Sprouts が後手必勝であることを図を使って説明しなさい.

問 2 Sprouts が無限に続かないことを証明しなさい. [ヒント: 1 つの点は高々 3 本の線にしかならない. それぞれの点が 3 つの命 (liberty) をもっていると考えると, 初期局面には全体で $3n$ 個の命がある.]

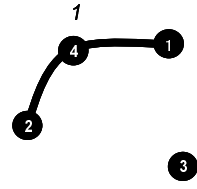
問 3 実は, Sprouts の局面 (position) を記述するには, 点の位置も線の形も捨象してよく, 線でどのように点がつながり, どのように領域 (region) が分割されるかの情報だけで十分である. ここでは以下の情報をプログラムで表現することにする.

- (a) 点の集合. 点には図 1 のように順に番号を振る.
- (b) 各点の残り命の数.
- (c) 各点の (線でつながった) 隣接点. 最大 3 つ.
- (d) 各点が面している領域. 最大 3 つ.
- (e) 領域の集合. 領域には図 1 のように適当に順に番号を振る.
- (f) 各領域の「成分」. 領域に面している単独の点, あるいは連結した点のグループを成分と呼ぶ. 例えば図 1(3) の領域 2 の成分は, 1,5 の連結グループ, および単独点 3 の 2 つであり, 領域 1 の成分は 1,2,4,5 の連結グループ 1 つである.

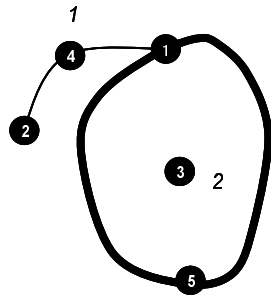
上記の (a) ~ (f) のデータ構造を設計し, 以下の小問に答えなさい. $n \leq 8$ とする. [ヒント: 点の集合および領域の集合は不定個数だが, 配列で表すのが便利である.]



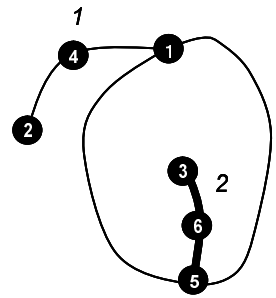
(1) 初期局面



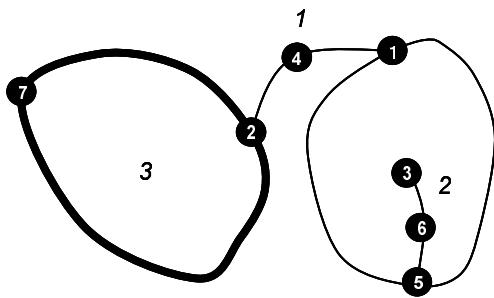
(2) 先手第1手



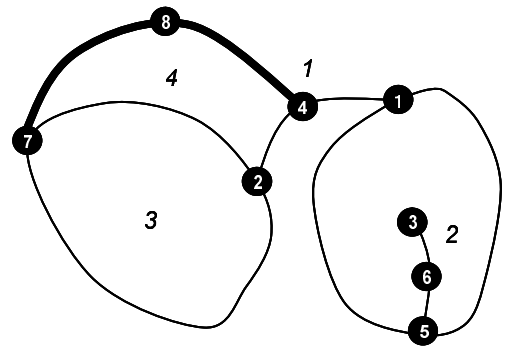
(3) 後手第1手



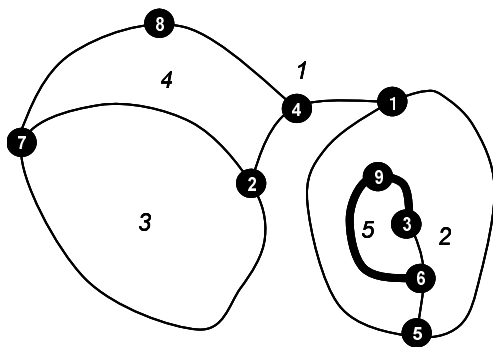
(4) 先手第2手



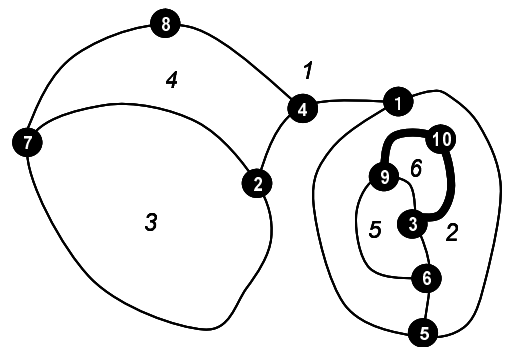
(5) 後手第2手



(6) 先手第3手

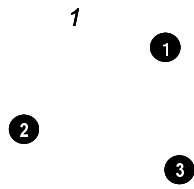


(7) 後手第3手

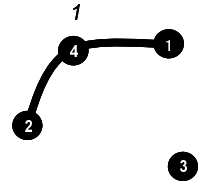


(8) 先手第4手—最終局面

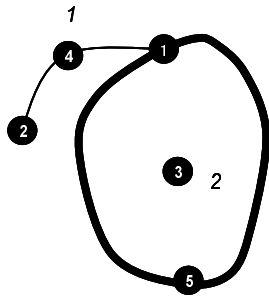
図1. Sproutsの進行例



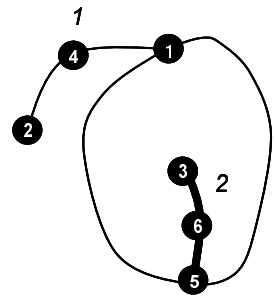
(1) 初期局面



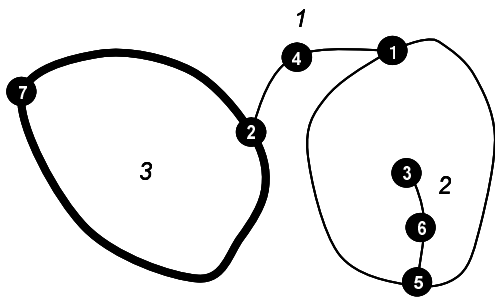
(2) 先手第1手



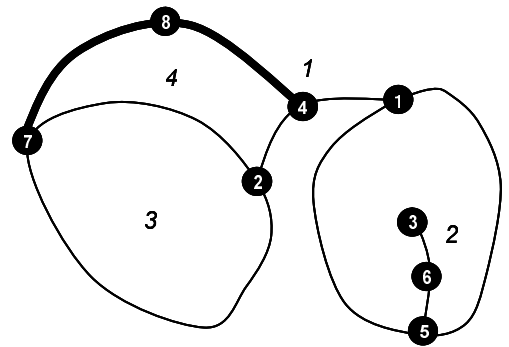
(3) 後手第1手



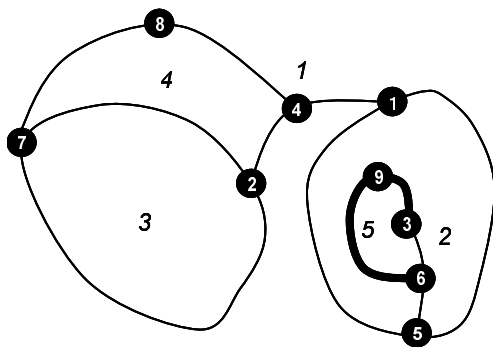
(4) 先手第2手



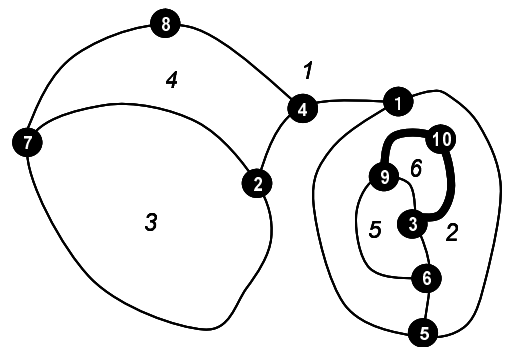
(5) 後手第2手



(6) 先手第3手



(7) 後手第3手



(8) 先手第4手—最終局面

図1 (再掲) . Sprouts の進行例

問 3(a) 図 1(3) の局面を表すデータ構造を生成するプログラムを書きなさい。別の局面の生成が容易に書けるように、この局面に固有の情報はプログラムになるべくまとめて定数で埋め込むか、設定ファイル等を書くようにしなさい。

問 3(b) 局面のデータ構造を文字列として観察するためのプログラムを書き、問 3(a) のデータ構造を出力しなさい。出力形式は任意であるが、口述試験で説明を求められることがある。

問 4 問 3 の局面のデータ構造には初期局面の点の数や先手・後手の区別が陽には含まれていない。局面のデータ構造から初期局面の点の数を計算によって求めて出力するプログラムを書き、問 3(a) で生成したデータ構造に適用し、出力が 3 になることを確認しなさい。

問 5 与えられた局面において、どの点とどの点を結ぶ手があるかを調べたい。ここでは線をどのように引くかは考えない。例えば、図 1(3) では、結び得る点の対は、2-2, 2-4, 2-5, 3-3, 3-5, 4-5 の 6 つである。このうち領域を分割しないのは 3-5 のみである、与えられた局面に対して、結び得る点の対を以下の形式ですべて出力するプログラムを書き、以下の小問に答えなさい。

#(2,2) #(2,4) #(2,5) #(3,3) (3,5) #(4,5)

領域を分割する手は対のカッコの前に記号 # をつけること。点の対 (x, y) は記号 # の有無にかかわらず、

$(x, y) < (x', y')$ if $x < x'$ or $(x = x'$ and $y < y')$

の順序で昇順に並べ、 (x, y) の対称解 (y, x) は出力しないこと。以下の小問では、局面のデータ構造を、問 3(a) と同様、プログラムで生成しなさい。

問 5(a) このプログラムを図 1(2) の局面に適用しなさい。

問 5(b) このプログラムを図 2 の局面に適用しなさい。

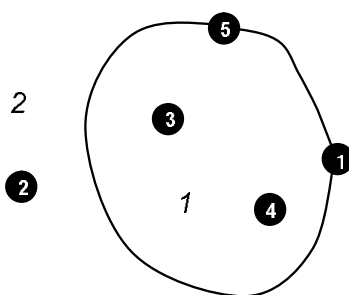


図 2 . 4 点 Sprouts の 1 つの局面

2006

Department of Creative Informatics
Graduate School of Information Science and Technology
The University of Tokyo

Programming

INSTRUCTIONS

1. Do not open this problem brochure until the signal to begin is given.
2. Write your examinee ID below on this cover.
3. An answer sheet and a draft sheet are delivered. Write down your examinee ID on the answer sheet.
4. The problem is a little long but it is carefully written to guide your answering. You are recommended to read the problem through before answering any question.
5. Write your programs on your own PC. **Write your answer, however, on your answer sheet for Question 1 and 2.** You can write part of your answer on the answer sheet if it is appropriate for other questions; probably, in case that you want to draw some figures by hand.
6. You may choose your favorite programming language.
7. You can consult only one printed manual of the programming language.
8. At the examination end, a USB memory is delivered to each examinee. Make a directory/folder whose name is the same as your examinee ID, and put your program file and relevant files under the directory/folder. Copy the directory/folder into the USB memory.
9. Then, the USB memory, answer sheet, and draft sheet are collected.
10. Stay at your seat after the collection, until all examinee results have been checked briefly.
11. After the brief check, retain your program execution environment on the PC so as to be able to resume it as soon as possible at the oral examination.
12. Bring your PC and this brochure together to the specified room for the oral examination and leave the room until you are called.

Examinee ID _____

The game Sprouts was invented on February 21 in 1967 by John H. Conway who is also the inventor of the Life game, and his graduate student Michael S. Paterson after some talk pursuing a simple and enjoyable game which can be played only with paper and pencil.

The rule of n -spot Sprouts is simple as follows. Write n small spots as the initial position. The players take turns moving. The player who cannot move will lose the game.

A move consists of two parts: draw a line connecting two spots, or a spot and itself, and then place a new spot on this new line. The following conditions must be obeyed.

- (1) Line may have any shape, but it cannot cross or touch any other existing spot or any line including itself.
- (2) No spot can have more than three lines attached to it. Note that a newly placed spot has two lines attached at its birth.

Fig. 1 on the next page illustrates a 3-spot Sprouts game, in which the first player wins at the final position Fig. 1(8). Spots in the initial position are numbered 1, 2, 3 and the spots placed by each player's moves are numbered one after another from 4. Either one of a divided region is given a new number sequentially as shown in Fig. 1.

Q1 Explain with illustrations that the second player will win 1-spot Sprouts.

Q2 Prove that Sprouts never goes forever. [Hints: We can say a spot has three *liberties* because it cannot be attached by more than three lines. So the initial position of n -spot Sprouts has totally $3n$ liberties.]

Q3 You can easily see that the exact coordinates of spots and the shapes of lines are not significant, but logical information about how spots are connected by lines and how regions are separated by lines are sufficient to represent any Sprouts position.

Take into account the following information to represent a Sprouts position, and assume $n \leq 8$.

- (a) the set of spots numbered as illustrated in Fig. 1.
- (b) liberties of each spot.
- (c) adjacent spots connected to each spot (at maximum, 3 spots).
- (d) regions faced by each spot (at maximum, 3 regions).
- (e) the set of regions numbered as illustrated in Fig. 1.
- (f) *components* of each region. We call a single spot or connected group of spots that faces a region a component of the region. For example, there are two components in region 2 in Fig. 1(3): a single spot 3 and a connected group of 1 and 5. And there is only one component in region 1: a connected group of 1, 2, 4, and 5.

Design a data structure for (a), ..., (f) above. [Hints: It is convenient to use arrays for the sets of spots and regions, in spite of their variable size.]

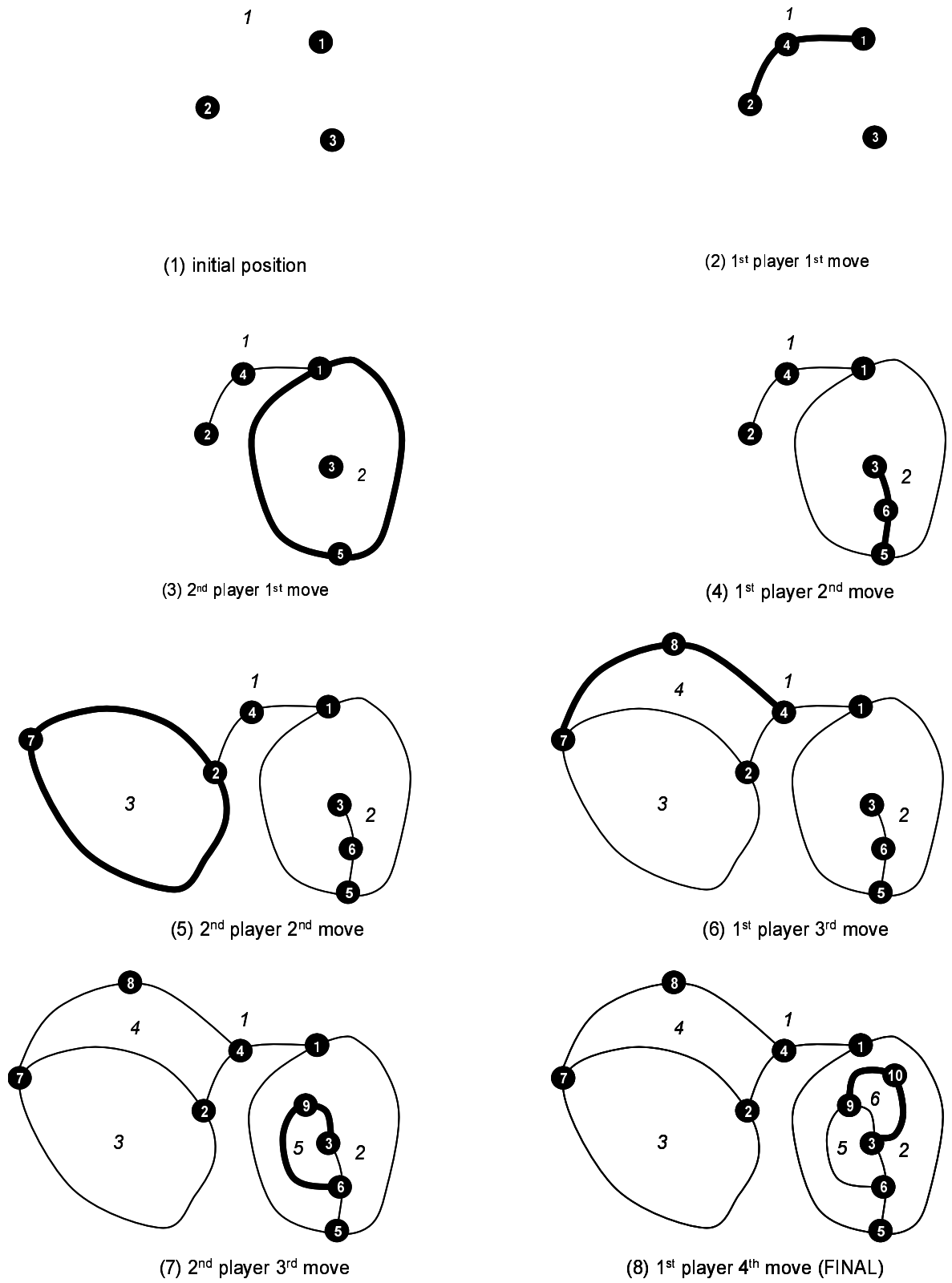
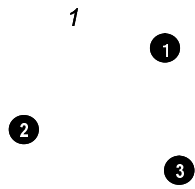
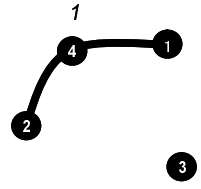


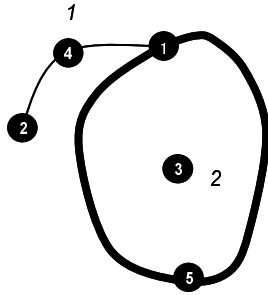
Figure 1. A Sample Game of 3-spot Sprouts



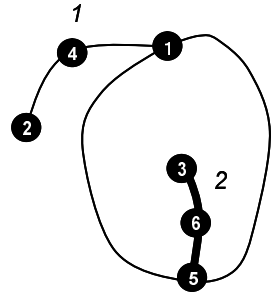
(1) initial position



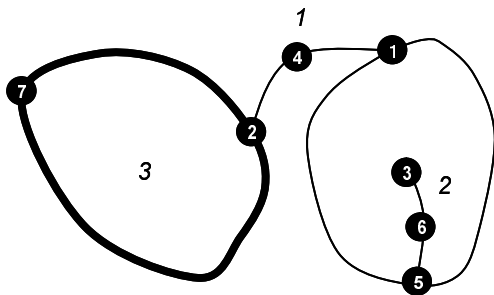
(2) 1st player 1st move



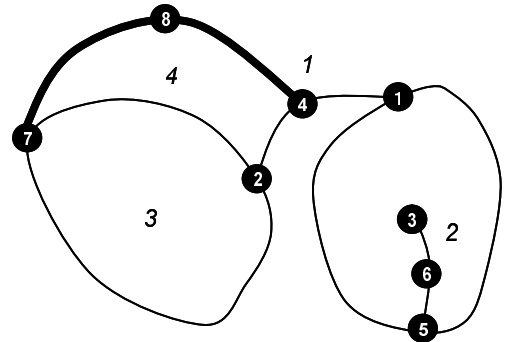
(3) 2nd player 1st move



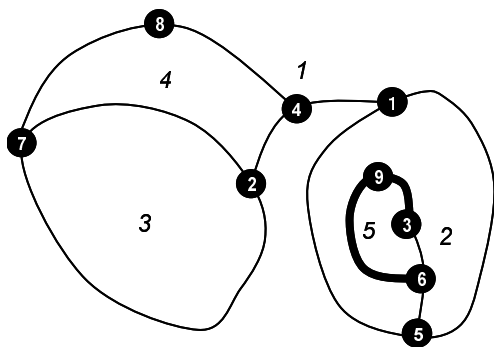
(4) 1st player 2nd move



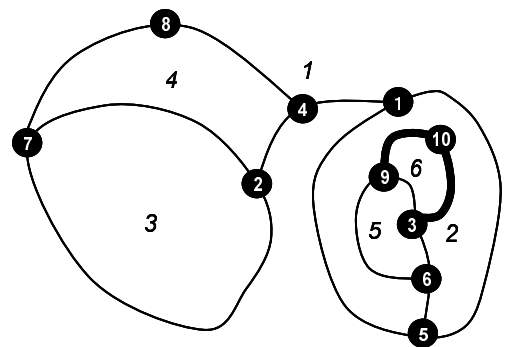
(5) 2nd player 2nd move



(6) 1st player 3rd move



(7) 2nd player 3rd move



(8) 1st player 4th move (FINAL)

Figure 1 (again). A Sample Game of 3-spot Sprouts

Q3(a) Write a program that makes the data structure for the position of Fig. 1(3). Information specific to this position should be embedded neatly in the program, or described in a configuration file or the like; that is, you should describe it in a way that can be easily made use of for other positions.

Q3(b) Write a program by which you can inspect the data structure by CUI (character user interface), and apply it to the position made in Q3(a). The design of the output format is left to you, but you will be required to explain it at the oral examination.

Q4 There is no explicit information about the initial number of spots and player's turn in the data structure made in Q3. However, these can be calculated from any Sprouts position. Write a program that outputs the initial number of spots from a given Sprouts position. Apply it to the position made in Q3(a) and be sure the output is 3.

Q5 We want to enumerate all possibilities which spots can be connected by the next move in a given Sprouts position. We do not take into account how a line is drawn for the present. For example, there are six connectable spot pairs 2-2, 2-4, 2-5, 3-3, 3-5, and 4-5 in the position shown in Fig. 1(3), out of which only 3-5 does not divide a region.

Write a program that prints all connectable spot pairs in the following format:

#(2,2) #(2,4) #(2,5) #(3,3) (3,5) #(4,5)

where every spot pair that divides a region are prefixed by a # symbol. Pairs should be sorted in ascending order by the following order relation:

$$(x,y) < (x',y') \quad \text{if } x < x' \text{ or } (x = x' \text{ and } y < y')$$

and the symmetric spot pair (y,x) of (x,y) should not be printed. In the following questions, make the data structure of the specified position in the same way you devised in Q3(a).

Q5(a) Apply the program to the position depicted in Fig. 1(2).

Q5(b) Apply the program to the position depicted in Fig. 2.

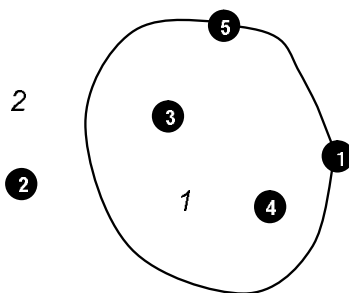


Figure 2. A position of 4-spot Sprouts