Application of Domino's Tiling Theory to calculate the probability of tatami mat arrangements in rectangular rooms, the number of mats required, and the areas where a full sheet of tatami mat cannot be arranged

Institute: Samsenwittayalai School | Thailand

Supervisor

Mr. Suthiphan Hiranpruk

Team member 1

Mr. Tinnapat Sittisuwan

Team member 2

Mr. Saran Laiwisetkun

Team member 3

Mr. Saknarin Srima

Table of contents

Introduction

- Domino Tiling is the process of tiling dominoes to perfection in the square or rectangular area.
- Tatami is an ancient mat of Japan used as a floor covering.
- We found that the play of the dominoes resembles the process of placing Tatami mats in Japanese. Then we use the method of Domino Tiling to solve the process of placing the tatami mats.

Objective

1) the number of methods for arranging the Tatami tiles into the room size 2*n and 3*n
2) the total number of tiles used in the arrangement, and
3) the remaining room space that cannot be filled with full sheets of tatami mats

$$\frac{\sqrt{2.8}}{3+2^{+}}$$

02

Method

Process of Tatami's
calculation programming

Variables and formulas

Variables

- *r_w (room width)*
- *r_l (room length)*
- *t_w (tatami width)*
- *t_l (tatami length)*

Formulas

Room area formula

$$r_{area} = r_{width} \times r_{length}$$

Tatami area formula

$$t_{area} = t_{width} \times t_{length}$$

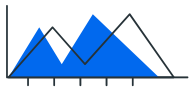
Program UI Design

```
------- Welcome to the tatami Calculator -------
Enter the room width (2 or 3) : 2
Enter the room length : 3
**--**--**--**--**
Enter number
1: Tatami Original Size
2: Custom Size
**--**--**--**--**
1
No. of Solution : 3
The number of No. of Tatami : 3
The number of Rest area in the room : 0.00 square meters
```

```
------- Welcome to the tatami Calculator -------
Enter the room width (2 or 3) : 3
Enter the room length : 4
**--**--**--**--**
Enter number
1: Tatami Original Size
2: Custom Size
**--**--**--**--**
1
No. of Solution : 11
The number of No. of Tatami : 6
The number of Rest area in the room : 0.00 square meters
```

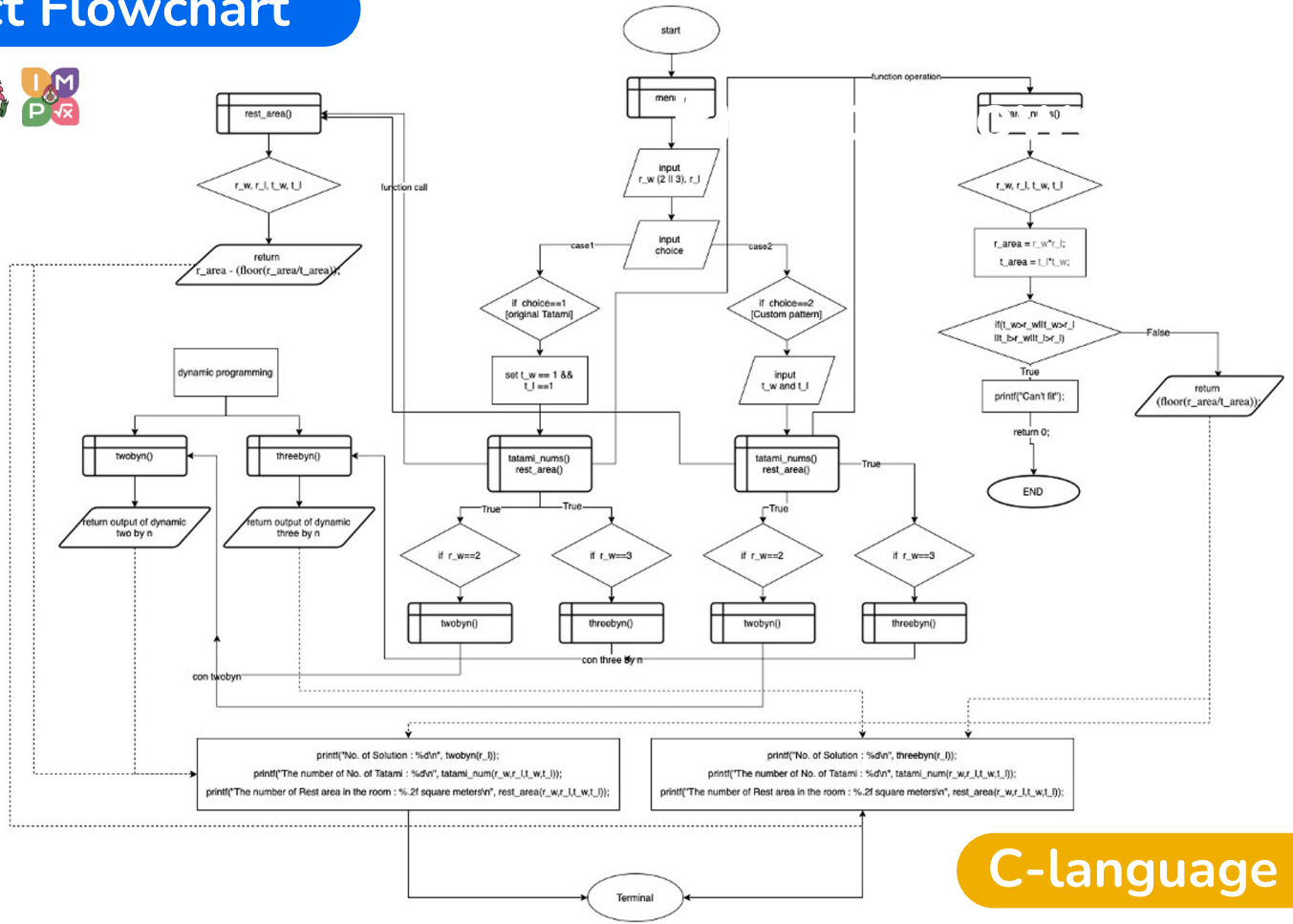
Code the program

C-language



INTEC GLOBAL

Project Flowchart



C-language program

```c
#include <stdio.h>
#include <math.h>
#include <string.h>

int r_area, t_area;
int r_w, r_l, t_w, t_l;
int x=1;
void menu(void);

int tatami_num(int r_w, int r_l, int t_w, int t_l){
    r_area = r_w*r_l;
    t_area = t_l*t_w;

    if(t_w>r_w||t_w>r_l||t_l>r_w||t_l>r_l){
        printf("Cant fit");
        return 0;
    }
    else return(floor(r_area/t_area));
}

float rest_area(int r_w, int r_l, int t_w, int t_l){
    return (r_area - t_area*floor(r_area/t_area));
}

int twobyn(int n){
    if (n <= 2)
      return n;
    return twobyn(n - 1) + twobyn(n - 2);
}

int threebyn(int n){
    int A[n + 1], B[n + 1];
    A[0] = 1, A[1] = 0, B[0] = 0, B[1] = 1;
    for (int i = 2; i <= n; i++) {
        A[i] = A[i - 2] + 2 * B[i - 1];
        B[i] = A[i - 1] + B[i - 2];
    }
    return A[n];
}

int main(){
    menu();
    if(r_w == 2){
        printf("No. of Solution : %d\n", twobyn(r_l));
    }
    if(r_w == 3){
        printf("No. of Solution : %d\n", threebyn(r_l));
    }
    printf("The number of No. of Tatami : %d\n", tatami_num(r_w,r_l,t_w,t_l));
    printf("The number of Rest area in the room : %.2f square meters\n",
rest_area(r_w,r_l,t_w,t_l));
    return 0;
}

void menu(void){
    int choice;
    printf("------ Welcome to the tatami Calculator ------\n");
    printf("Enter the room width (2 or 3) : ");
    scanf("%d", &r_w);
    printf("Enter the room length : ");
    scanf("%d", &r_l);
    printf("**--**--**--**--**--**\nEnter number\n1: Tatami Original Size\n2: Custom
Size\n**--**--**--**--**--**\n");
    scanf("%d", &choice);
    if(choice == 1){
        t_w = 1;
        t_l = 2;
    }
    if(choice == 2){
        printf("Enter the Tatami width : ");
        scanf("%d", &t_w);
        printf("Enter the Tatami width : ");
        scanf("%d", &t_l);
    }
}
```

Func. tatami_num

```c
int tatami_num(int r_w, int r_l, int t_w, int t_l){
    r_area = r_w*r_l;
    t_area = t_l*t_w;

    if(t_w>r_w||t_w>r_l||t_l>r_w||t_l>r_l){
        printf("Cant fit");
        return 0;
    }
    else return(floor(r_area/t_area));
```

```c
#include <stdio.h>
#include <math.h>
#include <string.h>

int r_area, t_area;
int r_w, r_l, t_w, t_l;
int x=1;
void menu(void);

int tatami_num(int r_w, int r_l, int t_w, int t_l){
    r_area = r_w*r_l;
    t_area = t_l*t_w;

    if(t_w>r_w||t_w>r_l||t_l>r_w||t_l>r_l){
        printf("Cant fit");
        return 0;
    }
    else return(floor(r_area/t_area));
}

float rest_area(int r_w, int r_l, int t_w, int t_l){
    return (r_area - t_area*floor(r_area/t_area));
}

int twobyn(int n){
    if (n <= 2)
      return n;
    return twobyn(n - 1) + twobyn(n - 2);
}

int threebyn(int n){
    int A[n + 1], B[n + 1];
    A[0] = 1, A[1] = 0, B[0] = 0, B[1] = 1;
    for (int i = 2; i <= n; i++) {
        A[i] = A[i - 2] + 2 * B[i - 1];
        B[i] = A[i - 1] + B[i - 2];
    }
    return A[n];
}

int main(){
    menu();
    if(r_w == 2){
        printf("No. of Solution : %d\n", twobyn(r_l));
    }
    if(r_w == 3){
        printf("No. of Solution : %d\n", threebyn(r_l));
    }
    printf("The number of No. of Tatami : %d\n", tatami_num(r_w,r_l,t_w,t_l));
    printf("The number of Rest area in the room : %.2f square meters\n",
rest_area(r_w,r_l,t_w,t_l));
    return 0;
}

void menu(void){
    int choice;
    printf("------ Welcome to the tatami Calculator ------\n");
    printf("Enter the room width (2 or 3) : ");
    scanf("%d", &r_w);
    printf("Enter the room length : ");
    scanf("%d", &r_l);
    printf("**--**--**--**--**--**\nEnter number\n1: Tatami Original Size\n2: Custom
Size\n**--**--**--**--**--**\n");
    scanf("%d", &choice);
    if(choice == 1){
        t_w = 1;
        t_l = 2;
    }
    if(choice == 2){
        printf("Enter the Tatami width : ");
        scanf("%d", &t_w);
        printf("Enter the Tatami width : ");
        scanf("%d", &t_l);
    }
}
```

Func. rest_area

```c
float rest_area(int r_w, int r_l, int t_w, int t_l){
    return (r_area - t_area*floor(r_area/t_area));
}
    else return(floor(r_area/t_area));
```

Func. main

```c
int main(){
    menu();
    if(r_w == 2){
        printf("No. of Solution : %d\n", twobyn(r_l));
    }
    if(r_w == 3){
        printf("No. of Solution : %d\n", threebyn(r_l));
    }
    printf("The number of No. of Tatami : %d\n",
tatami_num(r_w,r_l,t_w,t_l));
    printf("The number of Rest area in the room : %.2f
square meters\n", rest_area(r_w,r_l,t_w,t_l));
    return 0;
}
```

```c
#include <stdio.h>
#include <math.h>
#include <string.h>

int r_area, t_area;
int r_w, r_l, t_w, t_l;
int x=1;
void menu(void);

int tatami_num(int r_w, int r_l, int t_w, int t_l){
    r_area = r_w*r_l;
    t_area = t_l*t_w;

    if(t_w>r_w||t_w>r_l||t_l>r_w||t_l>r_l){
        printf("Cant fit");
        return 0;
    }
    else return(floor(r_area/t_area));
}

float rest_area(int r_w, int r_l, int t_w, int t_l){
    return (r_area - t_area*floor(r_area/t_area));
}

int twobyn(int n){
    if (n <= 2)
      return n;
    return twobyn(n - 1) + twobyn(n - 2);
}

int threebyn(int n){
    int A[n + 1], B[n + 1];
    A[0] = 1, A[1] = 0, B[0] = 0, B[1] = 1;
    for (int i = 2; i <= n; i++) {
        A[i] = A[i - 2] + 2 * B[i - 1];
        B[i] = A[i - 1] + B[i - 2];
    }
    return A[n];
}

int main(){
    menu();
    if(r_w == 2){
        printf("No. of Solution : %d\n", twobyn(r_l));
    }
    if(r_w == 3){
        printf("No. of Solution : %d\n", threebyn(r_l));
    }
    printf("The number of No. of Tatami : %d\n", tatami_num(r_w,r_l,t_w,t_l));
    printf("The number of Rest area in the room : %.2f square meters\n",
rest_area(r_w,r_l,t_w,t_l));
    return 0;
}

void menu(void){
    int choice;
    printf("------ Welcome to the tatami Calculator ------\n");
    printf("Enter the room width (2 or 3) : ");
    scanf("%d", &r_w);
    printf("Enter the room length : ");
    scanf("%d", &r_l);
    printf("**--**--**--**--**--**\nEnter number\n1: Tatami Original Size\n2: Custom
Size\n**--**--**--**--**--**\n");
    scanf("%d", &choice);
    if(choice == 1){
        t_w = 1;
        t_l = 2;
    }
    if(choice == 2){
        printf("Enter the Tatami width : ");
        scanf("%d", &t_w);
        printf("Enter the Tatami width : ");
        scanf("%d", &t_l);
    }
}
```

Func. 2byN and 3byN

```c
int twobyn(int n){
    if (n <= 2)
      return n;
    return twobyn(n - 1) + twobyn(n - 2);
}

int threebyn(int n){
    int A[n + 1], B[n + 1];
    A[0] = 1, A[1] = 0, B[0] = 0, B[1] = 1;
    for (int i = 2; i <= n; i++) {
        A[i] = A[i - 2] + 2 * B[i - 1];
        B[i] = A[i - 1] + B[i - 2];
    }
    return A[n];
}
```

$$\frac{\sqrt{2.8}}{3+2^{+}}$$

03

Test and result

Test and result of Tatami's calculation programming

Test and result

r_w=2, r_l=2, original size

```
------- Welcome to the tatami Calculator -------
Enter the room width (2 or 3) : 2
Enter the room length : 2
**--**--**--**--**
Enter number
1: Tatami Original Size
2: Custom Size
**--**--**--**--**
1
No. of Solution : 2
The number of No. of Tatami : 2
The number of Rest area in the room : 0.00 square meters
```

Compared the result

- To compare the output from the program with the Domino tiling theorem.

$$A_n = 2^2\left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right]. \quad A_n = 2^2\left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right] = 2^2\left[\frac{1}{2}\right] = 2\ solutions$$

- To compare the output from the program with the general form of arrangement theory.

$$T^4_{(2,2)} = \prod_{i=1}^{2}\prod_{j=1}^{2} 4\left[\cos^2\left(\frac{i\pi}{3}\right) + \cos^2\left(\frac{j\pi}{3}\right)\right] = 2\ solutions$$

- to compare the output from the program with Writing all possible cases

$2\ solutions$

Test and result

```
------- Welcome to the tatami Calculator -------
Enter the room width (2 or 3) : 2
Enter the room length : 4
**--**--**--**--**
Enter number
1: Tatami Original Size
2: Custom Size
**--**--**--**--**
1
No. of Solution : 5
The number of No. of Tatami : 4
The number of Rest area in the room : 0.00 square meters
```

```
------- Welcome to the tatami Calculator -------
Enter the room width (2 or 3) : 2
Enter the room length : 2
**--**--**--**--**
Enter number
1: Tatami Original Size
2: Custom Size
**--**--**--**--**
1
No. of Solution : 2
The number of No. of Tatami : 2
The number of Rest area in the room : 0.00 square meters
```

```
------- Welcome to the tatami Calculator -------
Enter the room width (2 or 3) : 3
Enter the room length : 4
**--**--**--**--**
Enter number
1: Tatami Original Size
2: Custom Size
**--**--**--**--**
1
No. of Solution : 11
The number of No. of Tatami : 6
The number of Rest area in the room : 0.00 square meters
```

```
------- Welcome to the tatami Calculator -------
Enter the room width (2 or 3) : 2
Enter the room length : 2
**--**--**--**--**
Enter number
1: Tatami Original Size
2: Custom Size
**--**--**--**--**
1
No. of Solution : 2
The number of No. of Tatami : 2
The number of Rest area in the room : 0.00 square meters
```

$\frac{\sqrt{2.8}}{3+2^{+}}$

04

Conclusion

- $A_n = 2^{2n^2} \prod_{i=1}^{n} \prod_{j=1}^{n} \left[\cos^2\left(\frac{i\pi}{2n+1}\right) + \cos^2\left(\frac{j\pi}{2n+1}\right)\right]$
- $T^4_{(m,n)} = \prod_{i=1}^{n} \prod_{j=1}^{m} 4\left[\cos^2\left(\frac{i\pi}{n+1}\right) + \cos^2\left(\frac{j\pi}{m+1}\right)\right]$
  - list all probability counts of patterns.

Suggestion

- The cost of arranged tiles in one room can be calculated by the number of pieces of tile.
- This program can be applied in industry. Example Tiles factory
- This information can be used in other research studies.
- This project can be used as a model or as a case study in applying mathematics to our daily life.

THANKS!

Institute: Samsenwittayalai School | Thailand

References

- combinatorics - Dominos (2×1 on $2 \times n$ and on $3 \times 2n$). (n.d.). Mathematics Stack Exchange. Retrieved April 1, 2022, from https://math.stackexchange.com/questions/1317885/dominos-2-times-1-on-2-times-n-and-on-3-times-2n
- *Random Domino Tilings - Wolfram Demonstrations Project*. (n.d.). Demonstrations.wolfram.com. Retrieved April 2, 2022, from https://demonstrations.wolfram.com/RandomDominoTilings/
- Kessler, P. (2008). *Arranging Tatami Mats*. https://www.mechanicaldust.com/Documents/TatamiMats_01.pdf
- Ruskey, F., & Woodcock, J. (2009). Counting Fixed-Height Tatami Tilings. *The Electronic Journal of Combinatorics*, *16*(1). https://doi.org/10.37236/215
- Erickson, A., & Ruskey, F. (n.d.). *Domino Tatami Covering is NP-complete*. https://arxiv.org/pdf/1305.6669.pdf
- Klarner, D., & Pollack, J. (1980). Domino tilings of rectangles with fixed width. https://doi.org/10.1016/0012-365X(80)90098-9
- Audibert, P. (2010). *Mathematics for informatics and computer science*. Iste ; Hoboken, Nj.Borys, K. (n.d.). *DOMINO TILING*. Retrieved April 4, 2022, from http://math.uchicago.edu/~may/REU2015/REUPapers/Borys.pdf
- Rué, J. (n.d.). *Domino tilings of the Aztec Diamond*. Retrieved April 5, 2022, from https://upcommons.upc.edu/bitstream/handle/2117/87721/Snapshot-2015-016.pdf
- Kenyon, R. (2000). Conformal invariance of domino tiling. https://projecteuclid.org/journals/annals-of-probability/volume-28/issue-2/Conformal-invariance-of-domino-tiling/10.1214/aop/1019160260.full
- Aamand, A., Abrahamsen, M., Ahle, T., & Rasmussen, P. (2020). *Tiling with Squares and Packing Dominos in Polynomial Time*. https://thomasahle.com/papers/tiling.pdf
- *Recursive Functions Computability and Logic*. (n.d.). Retrieved April 7, 2022, from https://homepages.hass.rpi.edu/heuveb/Teaching/Logic/CompLogic/Web/Presentations/Recursive%20Functions.pdf
- Alhazov1', A., Morita1, K., & Iwamoto1, C. (n.d.). *A Note on Tatami Tilings*. Retrieved April 9, 2022, from https://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/1691-01.pdf
- Kamtue, S., & Yang, B. (n.d.). *DOMINO-TILING PROBLEM*. Retrieved April 11, 2022, from https://math.mit.edu/research/undergraduate/spur/documents/2013Kamtue.pdf