

# BÍ MẬT THUẬT TOÁN NGẪU NHIÊN



**DƯƠNG NGUYỄN THÁI BẢO**

Team leader @ Big-O Coding

# NỘI DUNG

1. Bản chất sự  
ngẫu nhiên

2. Tại sao phải quan tâm  
tới sự ngẫu nhiên?

3. Tạo yếu tố  
ngẫu nhiên

4. Thuật toán  
ngẫu nhiên

# NỘI DUNG

1. Bản chất sự  
ngẫu nhiên

2. Tại sao phải quan tâm  
tới sự ngẫu nhiên?

3. Tạo yếu tố  
ngẫu nhiên.

4. Thuật toán  
ngẫu nhiên.



## Ngẫu nhiên là gì?

→ Là sự thiếu khả năng đưa ra kết luận chính xác.



# NỘI DUNG

1. Bản chất sự  
ngẫu nhiên

2. Tại sao phải quan  
tâm tới sự ngẫu nhiên?

3. Tạo yếu tố  
ngẫu nhiên.

4. Thuật toán  
ngẫu nhiên.

Sự ngẫu  
nhiên ở  
khắp mọi  
nơi

---





# Tại sao phải quan tâm sự ngẫu nhiên?

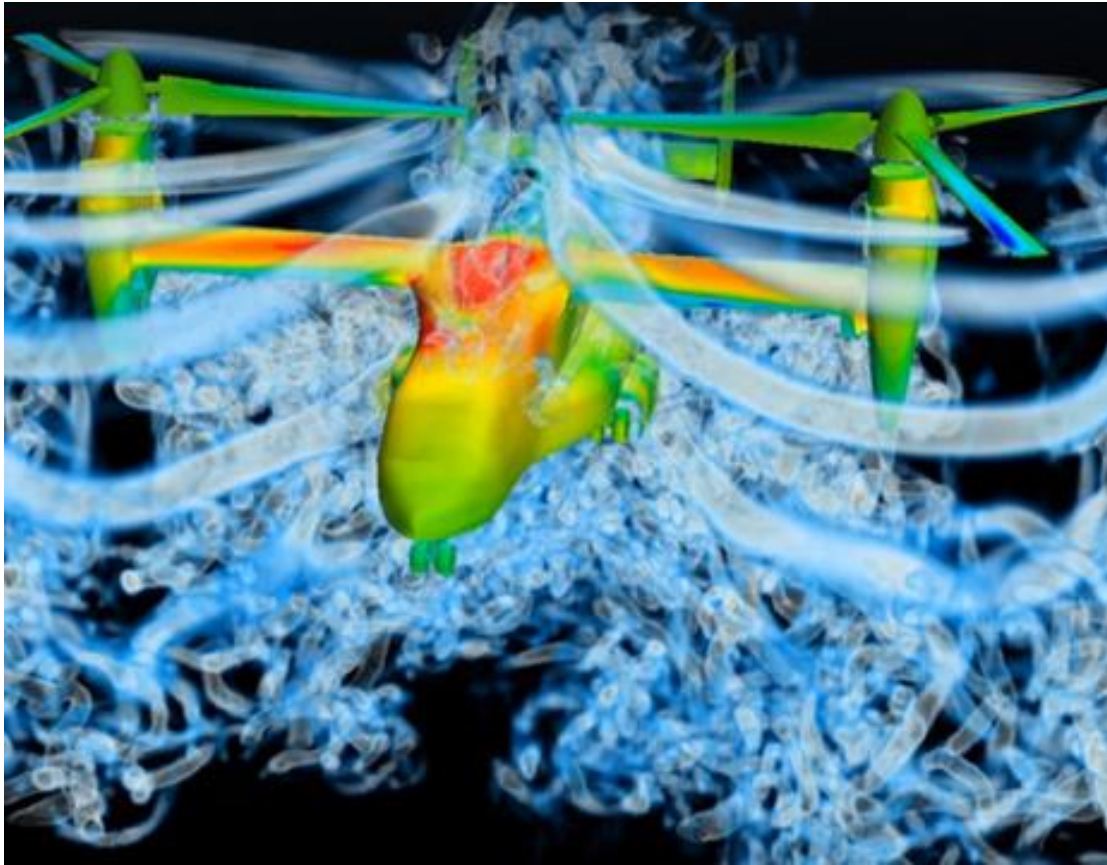
Bài bạc và cá cược



# Tại sao phải quan tâm sự ngẫu nhiên

Những chiếc “chìa khóa số” không  
được phép dễ đoán.





Mô phỏng vật lý phải  
sát với thực tế.

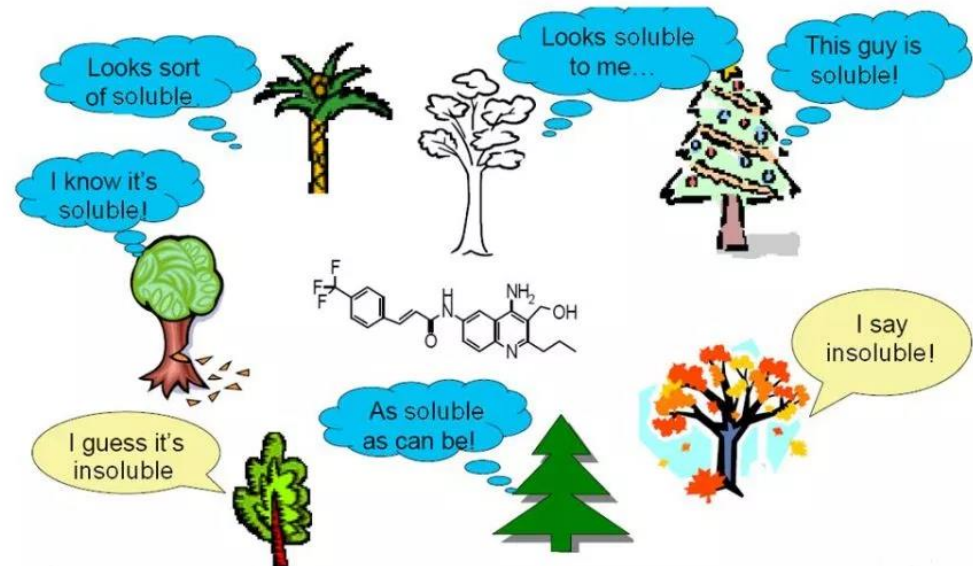
Tại sao phải quan tâm sự  
ngẫu nhiên?

# Tại sao phải quan tâm sự ngẫu nhiên?

- Khởi tạo giá trị ban đầu
- Lấy mẫu
- Thuật toán tối ưu
- Thuật toán Rừng ngẫu nhiên
- ...

## Random Forest

Machine Learning Method



# NỘI DUNG

1. Bản chất sự  
ngẫu nhiên

2. Tại sao phải quan tâm  
tới sự ngẫu nhiên?

3. Tạo yếu tố  
ngẫu nhiên.

4. Thuật toán  
ngẫu nhiên.

# Sự ngẫu nhiên có thực sự ngẫu nhiên?



**CTRL + C**



**CTRL + V**





Thời gian



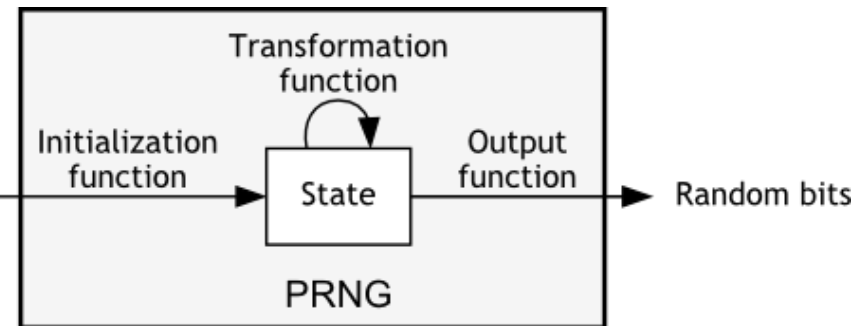
Nhiều khí quyển



LavaRand

# Tạo số ngẫu nhiên

Seed



Ví dụ: Linear Congruential Generator

```
1. int a = 3422, b = 13234, m = 28289;
2. int x = seed;
3. int random_number() {
4.     x = (a * x + b) % m;
5.     return x;
6. }
```

# NỘI DUNG

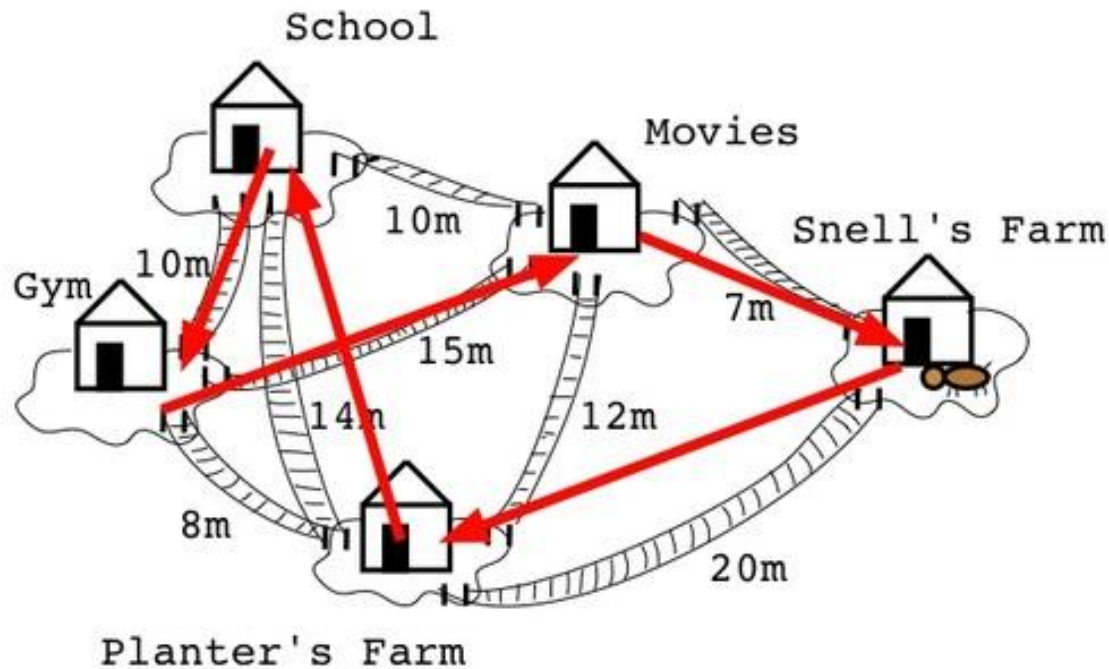
1. Bản chất sự  
ngẫu nhiên

2. Tại sao phải quan tâm  
tới sự ngẫu nhiên?

3. Tạo yếu tố  
ngẫu nhiên.

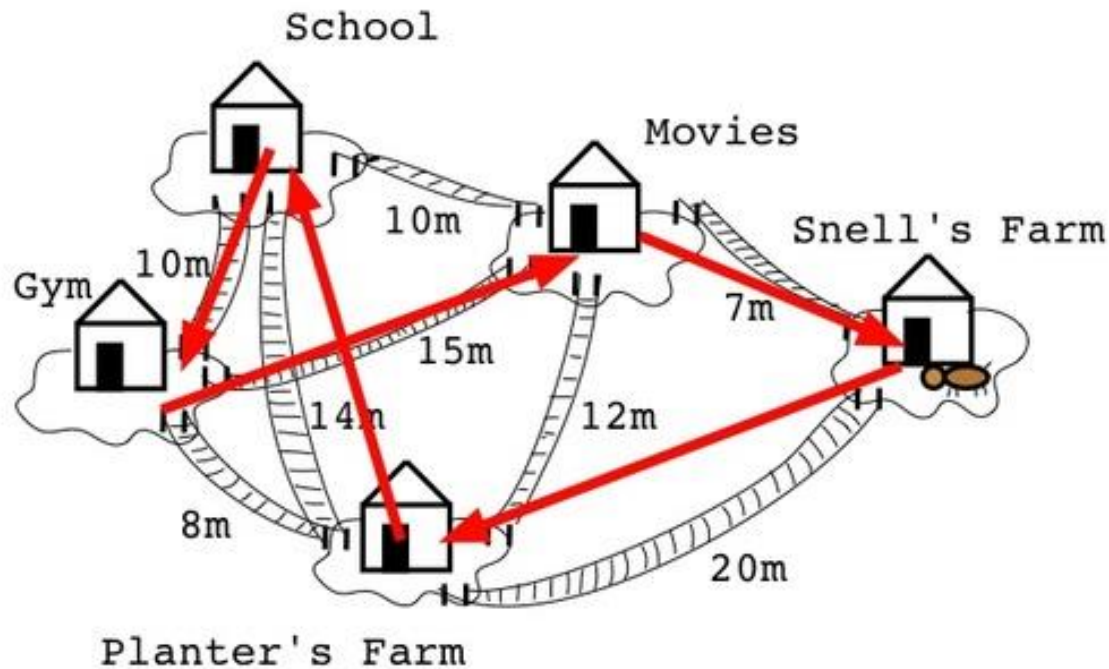
4. Thuật toán  
ngẫu nhiên.

# Bài toán người giao hàng (TSP)



- School → Planter's Farm → Snell's Farm → Movies → Gym → School = 66m
- Movies → Snell's Farm → Planter's Farm → Gym → School → Movies = 55m
- Gym → Movies → Snell's Farm → Planter's Farm → School → Gym = 66m
- ...

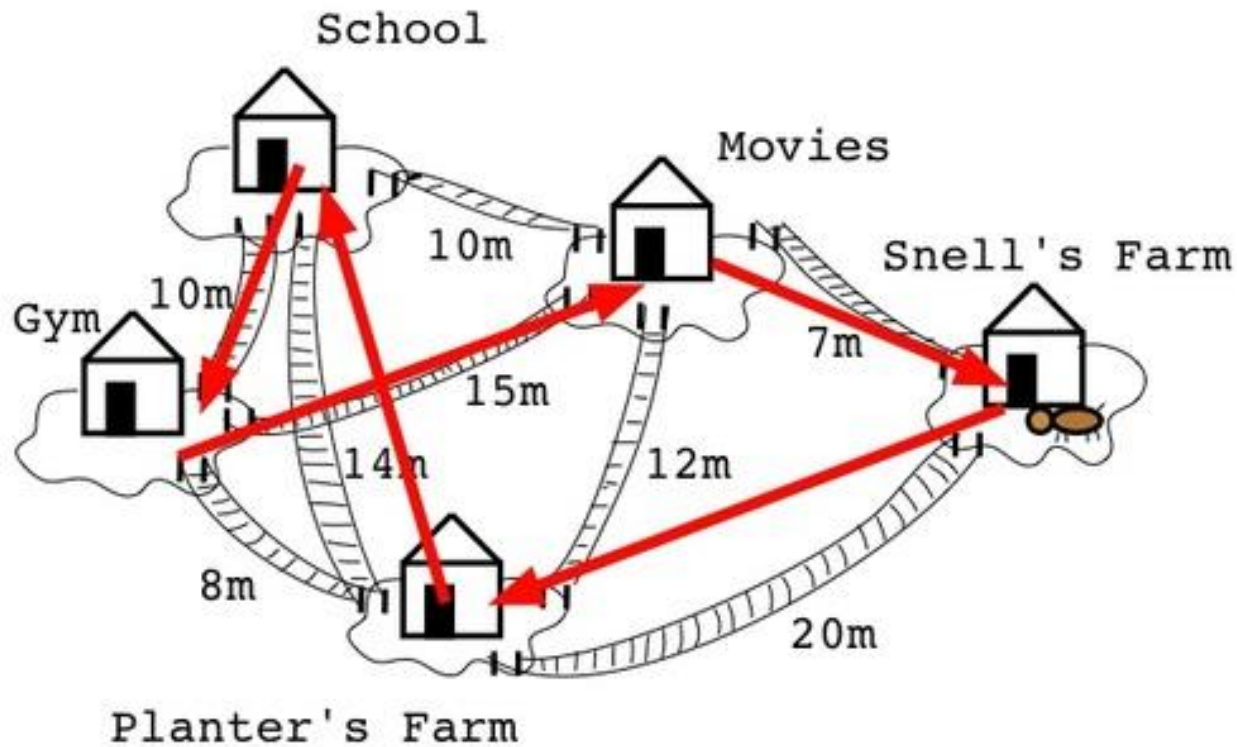
# Bài toán người giao hàng (TSP)



- $N$  ngôi nhà  $\rightarrow$  khoảng  $N!$  hành trình khác nhau.
- 100 ngôi nhà, thử được 1 hành trình/giây
- $\rightarrow 100!$  hành trình =  $2.96 \times 10^{148}$  **thế kỷ**

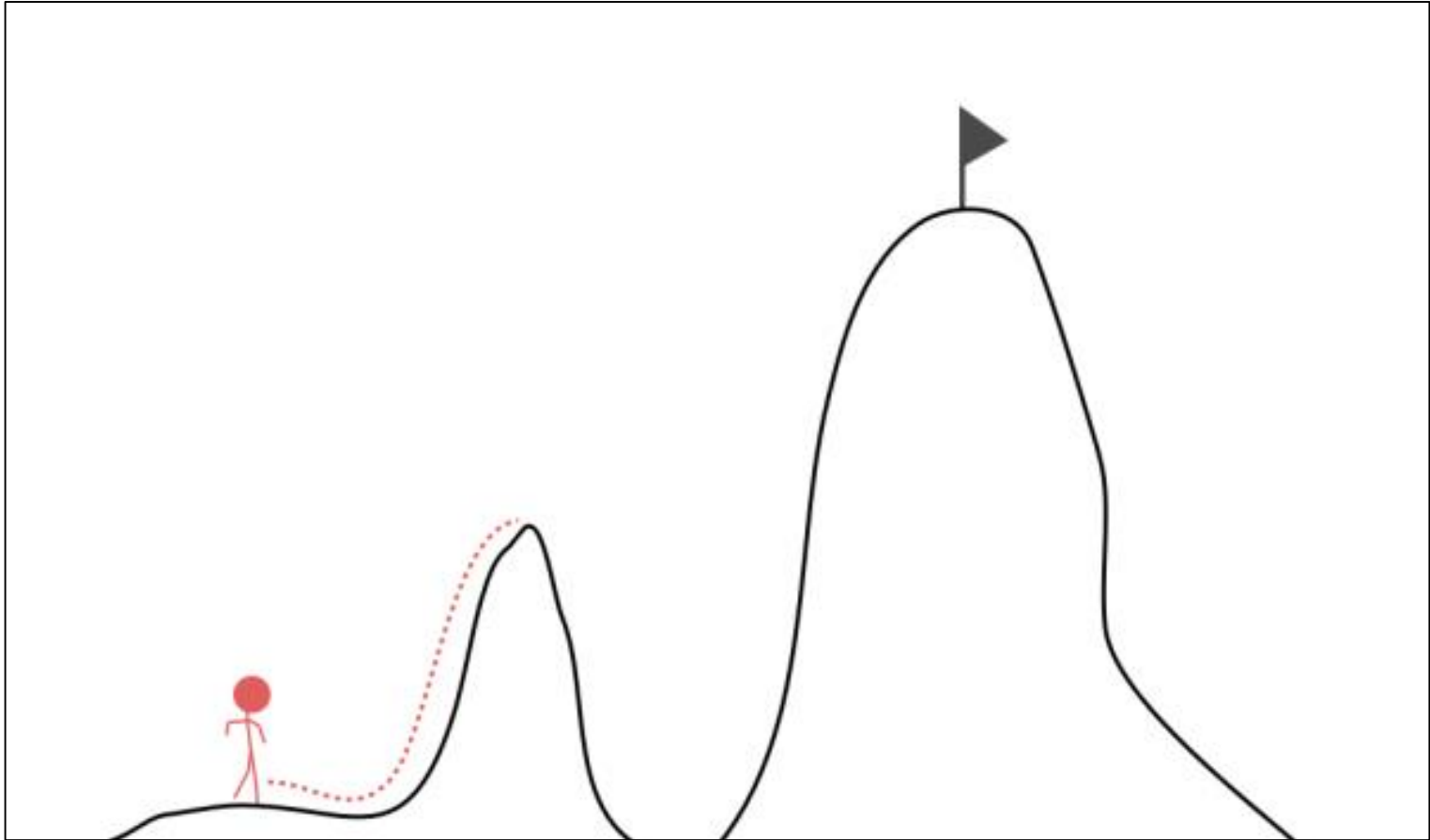


## Bài toán người giao hàng (TSP)



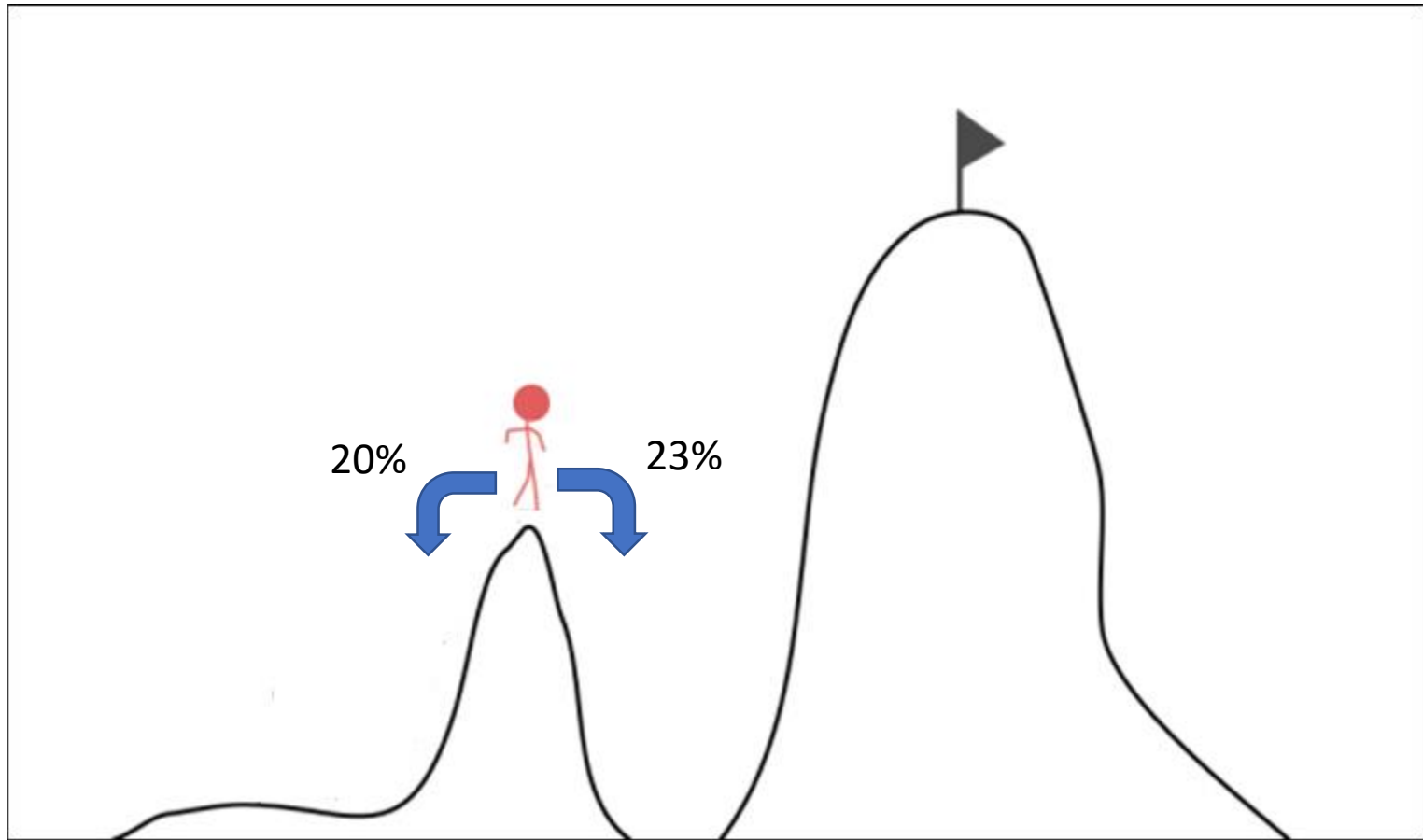
**Chưa có lời giải vừa nhanh vừa chính xác**

# Thuật toán Hill Climbing



→ Dừng ngay khi xung quanh không còn giá trị tốt hơn

# Simulated Annealing



→ Chấp nhận ra khỏi “vùng an toàn”

# Simulated Annealing

1. Bắt đầu với một vị trí *ngẫu nhiên*.
2. Ở mỗi bước:
  - Nếu một vị trí lân cận cho kết quả tốt hơn  
→ di chuyển tới đó.
  - Ngược lại → lựa chọn *ngẫu nhiên* nên ở lại hay đi tiếp.

→ Trên thực tế, hơn **90%** trường hợp Simulated Annealing cho kết quả tốt hơn Hill Climbing.

Link đọc: <http://www.psychicorigami.com/2007/06/28/tackling-the-travelling-salesman-problem-simulated-annealing/>

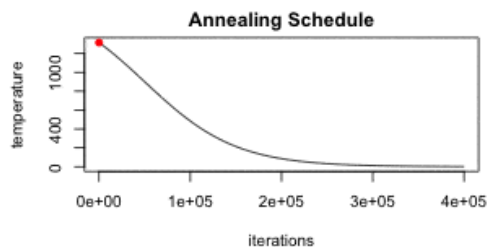


# TSP với Simulated Annealing

1. Bắt đầu với một *lộ trình ngẫu nhiên*.
2. Ở mỗi bước, đổi chỗ 2 ngôi nhà bất kì:
  - Nếu lộ trình này ngắn hơn → di chuyển tới lộ trình này.
  - Ngược lại → lựa chọn *ngẫu nhiên* nên ở lại hay đi tiếp.

# TSP với Simulated Annealing

Distance: 43,499 miles  
Temperature: 1,316  
Iterations: 0



# THANK YOU

**Dương Nguyễn Thái Bảo**

@company website: [bigocoding.com](http://bigocoding.com)

@email: [duongbao9998@gmail.com](mailto:duongbao9998@gmail.com)