

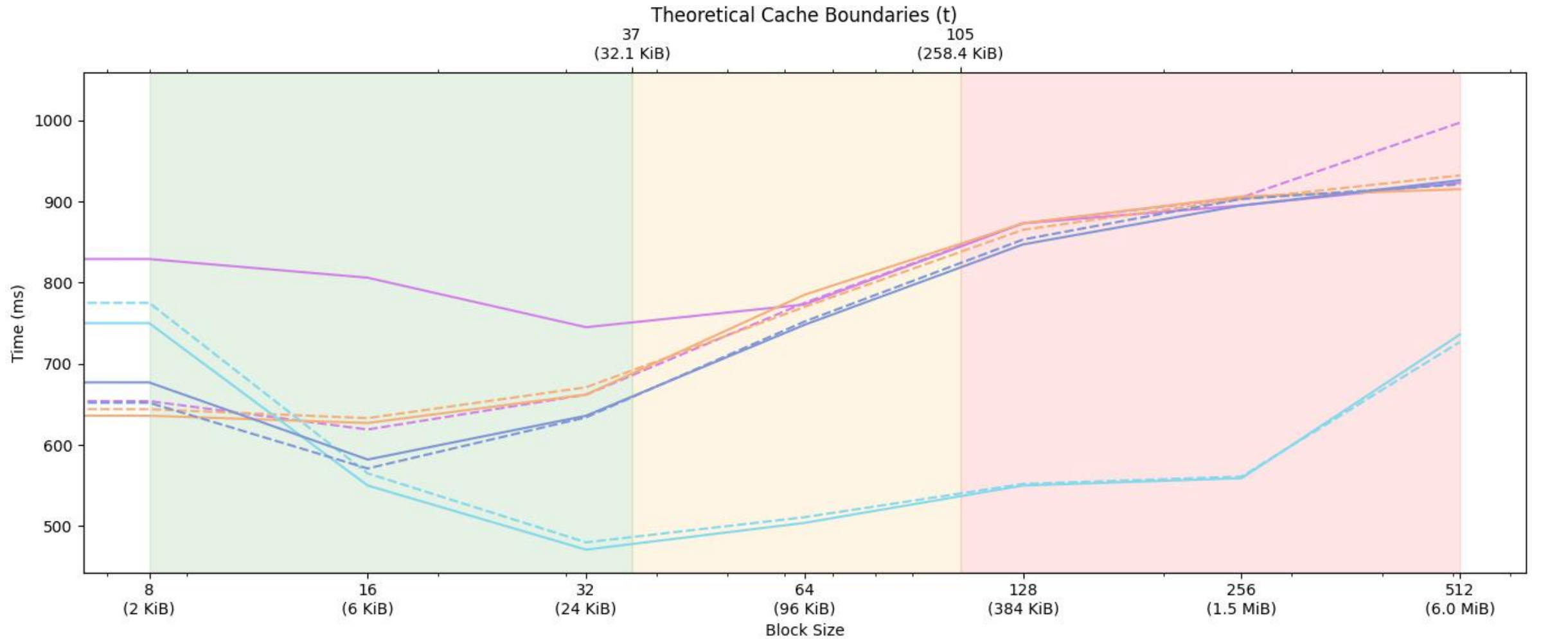
# **Aula 04** **Heurísticas e aleatoriedade**

Supercomputação



# **Comentários sobre a Atividade 2**

## Benchmark Results: Time vs. Block Size by Compiler and Optimization Level



Compilers   CLANG   GCC   INTEL icpx   NVIDIA nvc++   Optimization Flags   OO2   OO3   L1 Target   L2 Target   L3 Target

[Clique aqui para ter acesso ao relatório completo do Emil](#)

# Pipeline de Otimização

- Princípio da localidade temporal e espacial
- Hierarquia de memória (L1, L2, L3)
- Fatiamento dos dados em blocos para melhorar o aproveitamento da memória cache.
- Flags de Otimização



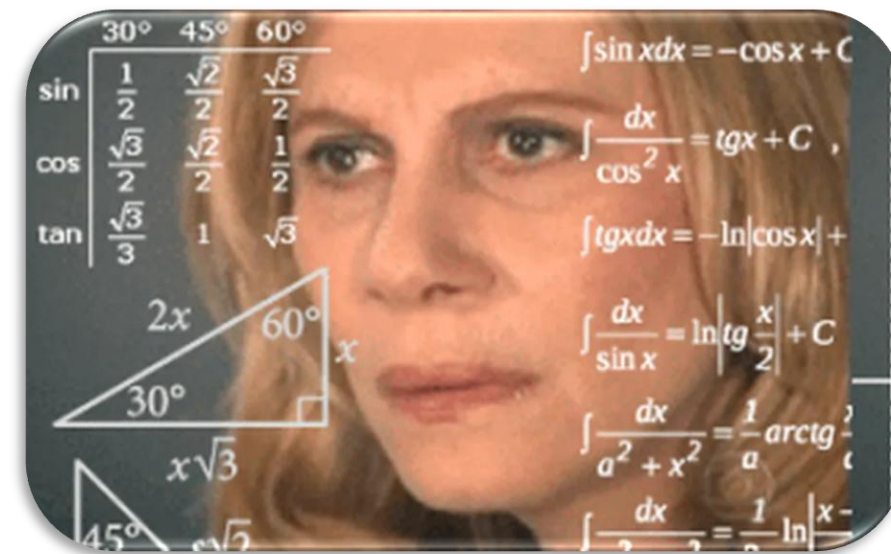
# Hora de olhar para a estratégia



# Lembra de heurística?

Uma heurística é uma regra usada para resolver problemas de forma mais rápida, mesmo que não garanta a melhor solução.

- **Busca em grafos:** usar Best First Search ou A\* que exploram caminhos mais promissores primeiro.
- **Algoritmos genéticos:** aplicar mutações aleatórias em soluções candidatas para escapar de mínimos locais.
- **Ordenação de dados:** Bubble Sort percorre várias vezes o vetor, trocando vizinhos fora de ordem.
- **Estratégias de busca:** exaustiva ou força bruta, testa todas as possibilidades até achar a solução, gulosa escolhe sempre a opção localmente melhor,  
**Aleatoriedade: em vez de percorrer em ordem, escolhe candidatos aleatórios.**



# Falando em Aleatoriedade....

- O que são valores aleatórios?
- Existe padrão nos valores aleatórios na computação?
- Como garantir aleatoriedade em intervalos de busca?





**BRACE YOURSELF**

**PRACTICE QUESTIONS ARE  
COMING**

makeameme.org