

Aula 6

Estrutura de Dados

Prof. Vinicius Pozzobon Borin

Conversa Inicial

- Investigaremos uma nova estrutura de dados
 - Grafos
- Como descobrir um grafo
 - Busca em largura
 - Busca em profundidade
- Caminho mínimo em grafo
 - Algoritmo de Dijkstra

Conceitos de grafos

O problema do mapeamento rodoviário



O problema do mapeamento rodoviário



Definição

- Grafo G é um conjunto de vértices conectados através de arestas sem uma distribuição fixa ou padronizada

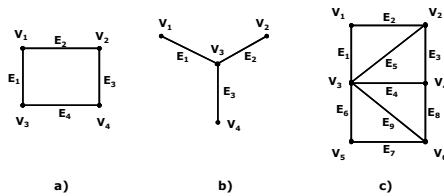
7

Definição

- Vértices V de um grafo são seus pontos. Cada ponto poderá ser um ponto de encontro entre caminhos (rotas) de um grafo, por exemplo
- Arestas E são as linhas de conexão entre vértices. Cada aresta conecta dois vértices

8

Exemplos



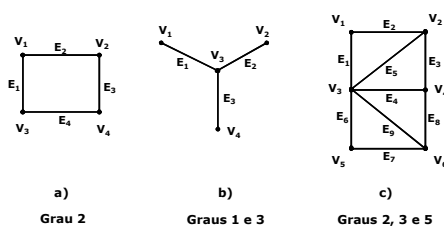
9

Conceitos

- Grafo completo
 - Quando existe uma, e somente uma, aresta para cada par distinto de vértices
- Grafo trivial
 - É um grafo com unicamente um vértice

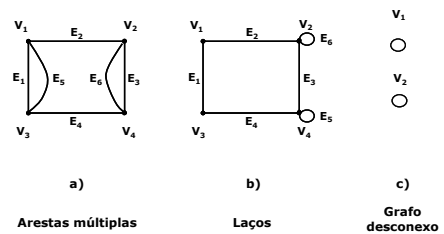
10

Grau de um grafo



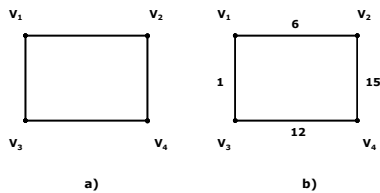
11

Particularidades em grafos



12

Grafo ponderado



Aplicações

- Encontrar rotas e melhores trajetos em mapas
- Escrever algoritmos de inteligência artificial que calculam o menor número de movimentos necessários para a vitória em uma partida de damas ou xadrez
- Mapear um jogo de tabuleiro para criar jogadas e movimentos

Aplicações

- Encontrar algo bastante próximo de você, como o médico mais próximo conveniado ao seu plano de saúde
- Conexões e tabelas de roteamento em redes de computadores
- Mapeamento de interações em redes sociais

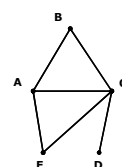
Representações de grafos

Matriz de adjacências

- Matriz de dimensão V , onde V é o número de vértices. E preenchida com

$\begin{cases} 0, \text{ caso o outro vértice não tenha conexão com o vértice analisado} \\ 1, \text{ caso o outro vértice tenha conexão com o vértice analisado} \end{cases}$

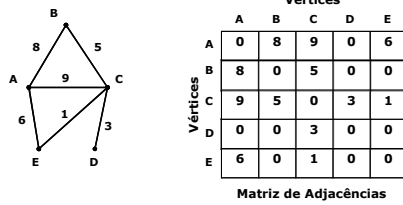
Matriz de adjacências



	Vértices				
	A	B	C	D	E
A	0	1	1	0	1
B	1	0	1	0	0
C	1	1	0	1	1
D	0	0	1	0	0
E	1	0	1	0	0

Matriz de Adjacências

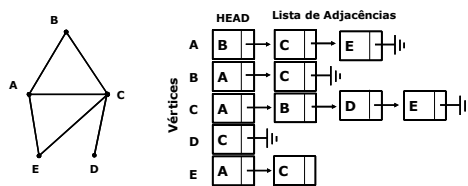
Matriz de adjacências com grafo ponderado



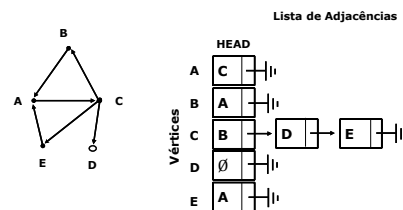
Lista de adjacências

- Criamos listas encadeadas de vizinhos de cada vértice
- Vizinhos de um vértice são todos os outros vértices que estão conectados a ele

Lista de adjacências



Lista de adjacências com grafo dirigido



Comparativo

- Em uma matriz de adjacência, sempre utilizaremos $|V|^2$ de espaço na memória, ou seja, se tivermos um grafo com 10 mil vértices, teríamos 100.000.000 bytes de uso de memória
- Portanto, uma desvantagem dessa representação é o excesso de uso de memória, especialmente para grafos grandes

Comparativo

- Em termos de complexidade de tempo de execução, a matriz é mais rápida que a lista de adjacência para grafos densos $|E|=|V|^2$, pois acessa qualquer dado com o mesmo tempo
- Por fim, essa representação é mais simples de ser aplicada para grafos ponderados

Comparativo

- Uma lista de adjacência é mais rápida, e emprega menos espaço de memória para grafos esparsos $|E| = |V|$
- Em contrapartida, para grafos bastante densos, são mais lentos do que a versão matricial, isso porque grafos densos tendem a formar grandes listas de vizinhos, que, quando representadas por listas encadeadas, geram um tempo de acesso ao dado bastante elevado

25

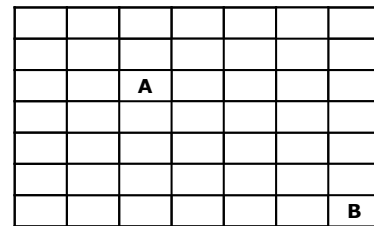
Busca em profundidade

26

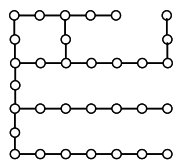
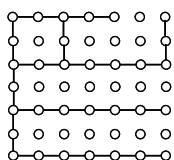
Busca em profundidade

- Conhecida como
 - *Depth-First Search (DFS)*
- Usamos para descobrir um grafo, ou fazer uma varredura, passando por todos os vértices uma única vez

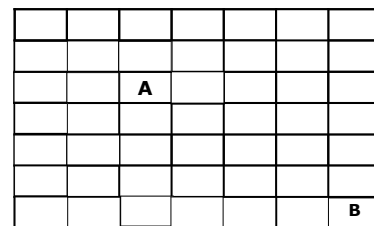
27



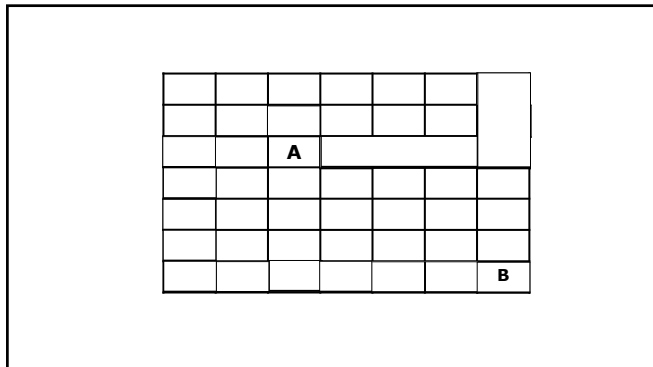
28



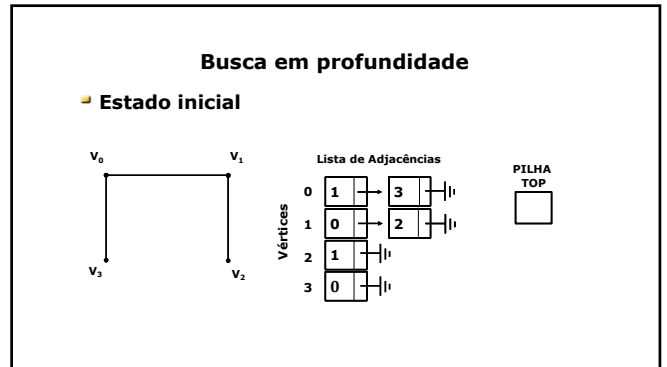
29



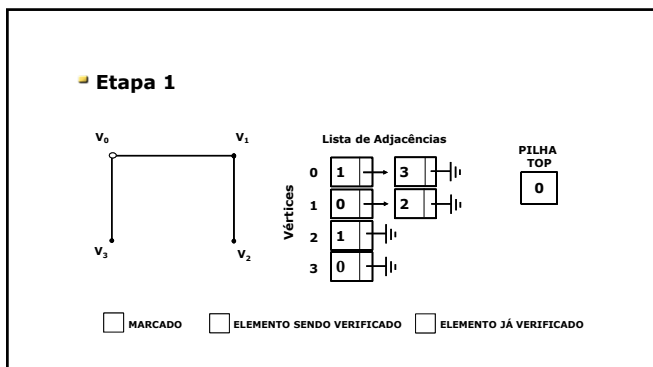
30



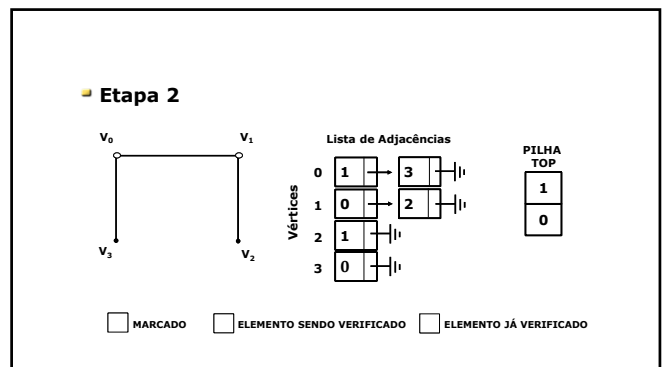
31



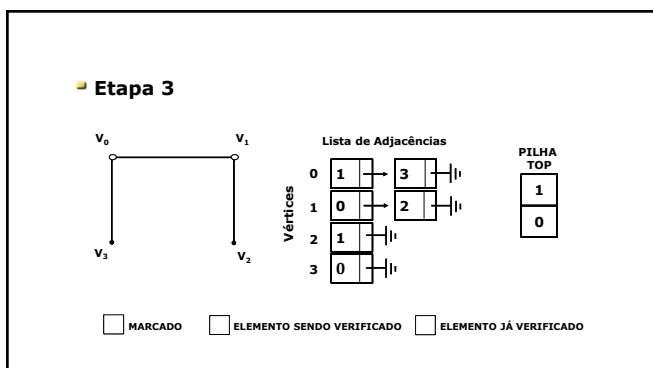
32



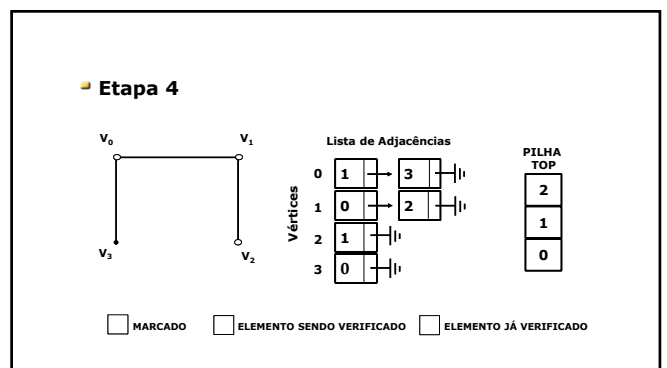
33



34

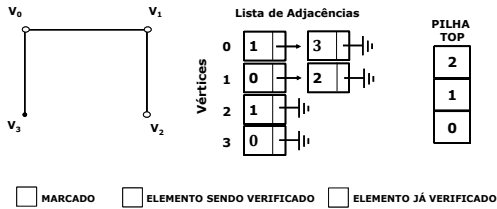


35



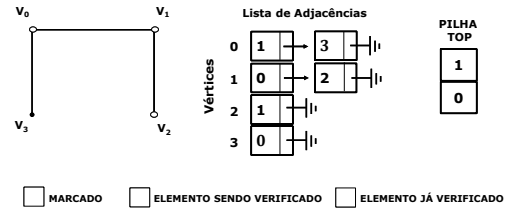
36

Etapa 5



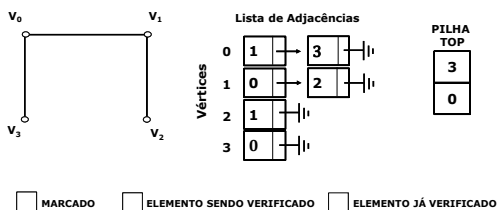
37

Etapa 6



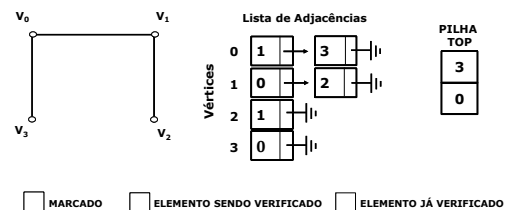
38

Etapa 7



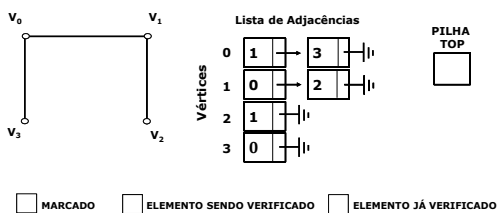
39

Etapa 8



40

Etapa 9



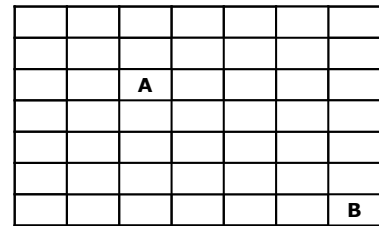
41

Busca em largura

42

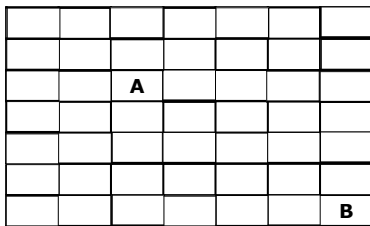
Busca em largura

- Conhecida como
 - Breath-First Search (BFS)*
- Usamos para descobrir um grafo, ou fazer uma varredura, passando por todos os vértices uma única vez



43

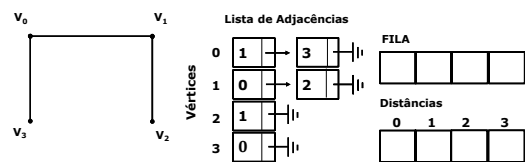
44



45

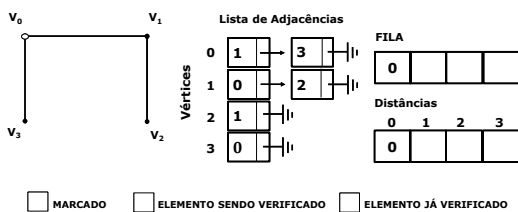
Busca em largura

Estado inicial



46

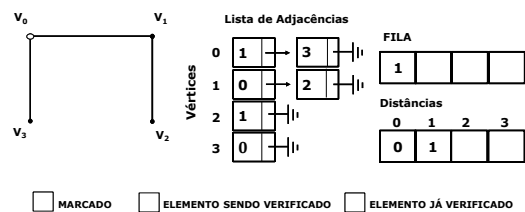
Etapa 1



☐ MARCADO
 ☐ ELEMENTO SENDO VERIFICADO
 ☐ ELEMENTO JÁ VERIFICADO

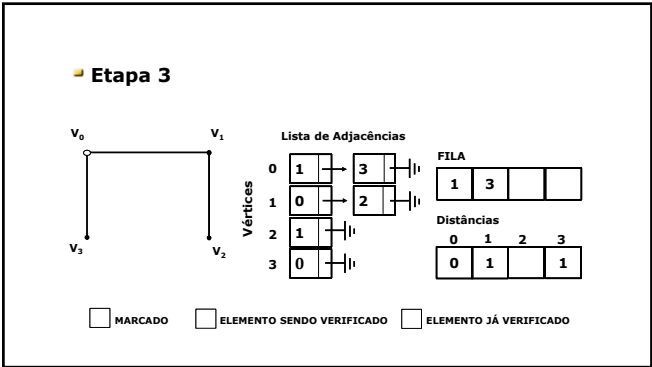
47

Etapa 2

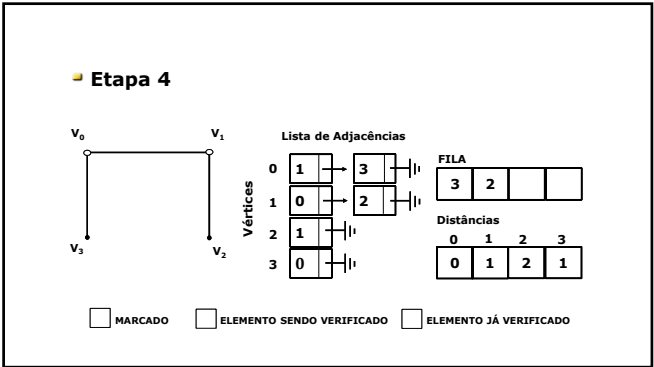


☐ MARCADO
 ☐ ELEMENTO SENDO VERIFICADO
 ☐ ELEMENTO JÁ VERIFICADO

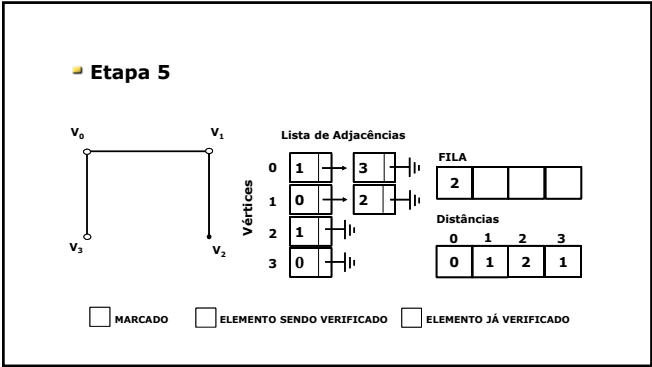
48



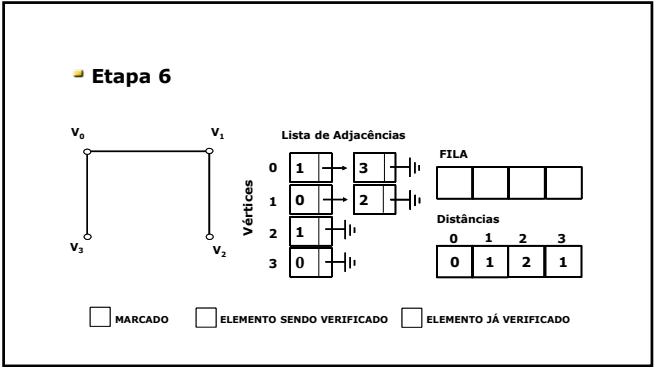
49



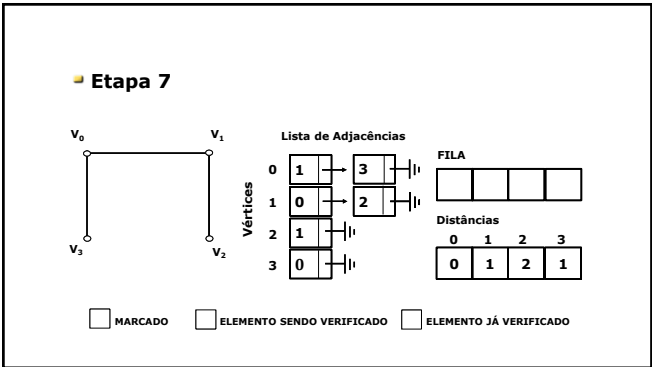
50



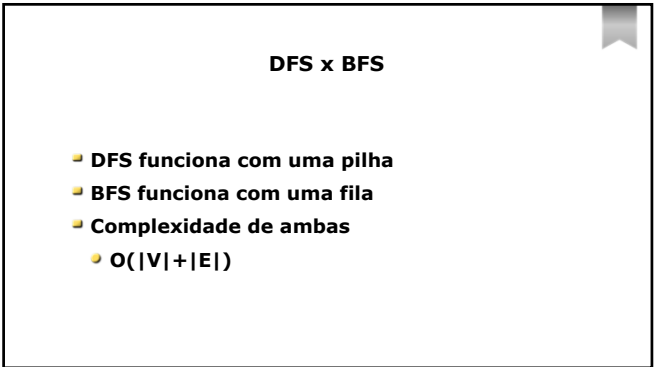
51



52



53



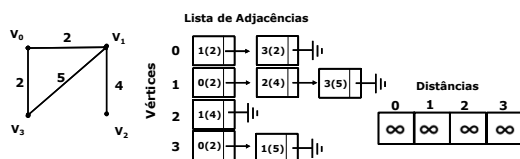
54

Caminho mínimo (algoritmo de Dijkstra)

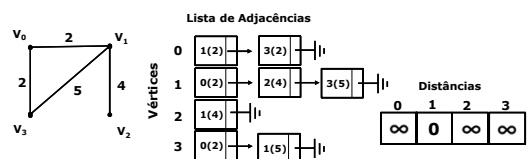
Algoritmo de Dijkstra

- Encontra a melhor rota/caminho entre dois vértices em um grafo ponderado
- Métrica aditiva
 - Menor rota considerando o menor peso somado entre os caminhos

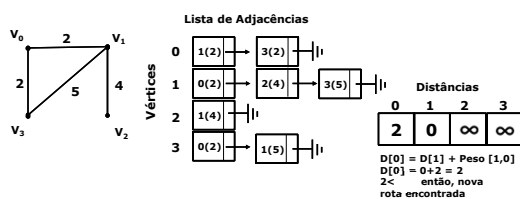
Estado inicial



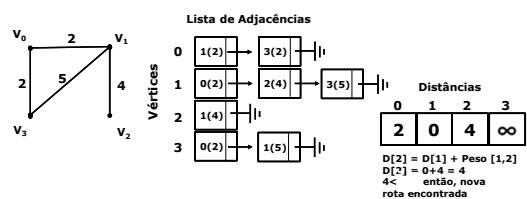
Etapa 1



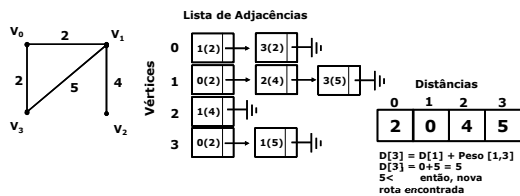
Etapa 2



Etapa 3



Etapa 4



61

Etapa 5



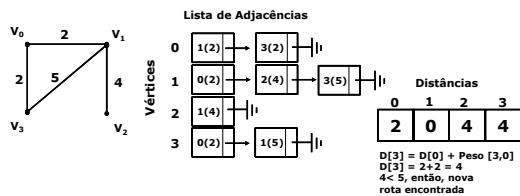
62

Etapa 6



63

Etapa 7



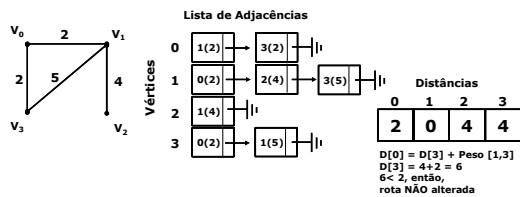
64

Etapa 8



65

Etapa 9



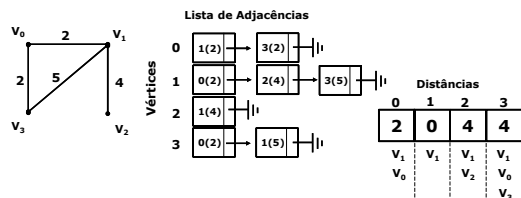
66

Etapa 10



67

Etapa 11



68

Referências

- KOFFMAN, E. B.; WOLFGANG, P. A. T. **Objetos, abstração, estrutura de dados e projeto usando C++**. Grupo GEN, 2008. 978-85-216-2780-7.
- Drozdek, A. **Estrutura de dados e algoritmos em C++**. Tradução da 4. ed. norte-americana. Cengage Learning Brasil, 2018.

69

70

- Bhargava, A. Y. **Entendendo algoritmos**. São Paulo: Novatec, 2017.
- ASCENCIO, A. F. G.; ARAÚJO, G. S. de. **Estruturas de dados: algoritmos, análise da complexidade e implementações em JAVA e C/C++**. São Paulo: Pearson Prentice Hall, 2010.

71

- FERRARI, R. et al. **Estruturas de dados com jogos**. São Paulo: Elsevier, 2014.
- CORMEN, Thomas H. **Algoritmos: teoria e prática**. 3. ed. São Paulo: Elsevier, 2012.

72