ТАБЛИЦА ДЕРЕВЬЕВ И ГРАФОВ 28 марра 2027. 123:03			
	binary tree	n days tree amorphisms rawwe nogeoper gale DAG class Block (chineta):	graphs adjacency_list = [[1, 2], [2, 3], [3], [1], [1] adjacency_distressis = (0: [1, 2], 1: [2, 3], 2: [3], 3: [])
	class Transmiss: defint(unit_, valet, laft=dinon, right=dinon):	class Mode (molymes); date mail_molsf_n_val=dinner, children=dinner); salf.childrens = childrens or [] date mail_molsf_n_val=dinner; salf.childrens = childrens or []	<pre>graph_melghad = fair files in 'err (), 'err (), 'err (err i, 'err 2), 'err (er edge=eff; 21, fb, 11, fb, 21, fb, 4]</pre>
	11, 12, 13, 14 = Trendicie(1), Trendicie(2), Trendicie(3), Trendicie(4), Trendicie(5) 11,1att, 11,right, 12,1att, 12,right = 12, 15, 17, 14	treei = Node(1, [Node(2, [Node(4), Node(5)]), Node(5, [Node(6), Node(7)]))	
preorder traversal	def interest (grainest[Trembed]): ***********************************	def (h(coat: 'Boke'); return (rest.ost) + (are for hid in root.okhidesa for are in de(hid)) if root wise ()	del fictione, media : [In (Marchaelland) Marchaelland Marchae
	<pre>def trowvest(rect: optional(treshole)): stack = reat and [rost] shile retain and [nost = exact.pup()]: stack = filter(films, (node.right, node.isf1)) reack = filter(films, (node.right, node.isf1))</pre>	def terrerei(men; "Gone") -> Limi((min)) tatik - men def (men) shiri - men def (men) print - men de (men) print - men de (men) stack += medic.childmen[1:-1]	Set traversippoint tital; this was "Blo B" this post Book " a "sand-popul"
level order traversal	def levelOrderFront; Optional[TransInde]): The state of	def treversi(rost: "Bode"); level - rost and [rost] simple [avaid [avaid for an in level] level = [this for mode in level for kid in code.children)	of traversalgraph); lent, see = [0]; [0] million of the control of
	of followed; entern [[n.val for a in level]] + for[hid for node in level for hid in (node.left, node.right) [[hid]) if level size []	def followed: return [[s.val for a in level]] + fo([bid for mode in level for bid in mode.children)] if level size []	def follows-[0],
	of transat([port.qrional[TeadBods)]); ques * collection.cppu([codt)) if post else line shiring to the collection.cppu([codt)) code ques attachist([libr(line, (code.lafe, code.riph))) ques attachist([libr(line, (code.lafe, code.riph)))	det transcentionat: "Ende"): quese rent out circlattion.organ([cont)) thin Spars and from a quest opposit([)): quest as mode that the proper opposition of the property of t	def Licentral (proph): William (Licentral Control Con
get kurts	def Salvest Optional(Province)). resons defront-latts + defront-right) or [rest.val] if rest also []	der folgelf, root, 'Mode'); return [ser for kid in root, shildren for err in ga(kid)] or [root, val]	
Horder traversal	<pre>def traveral(rost: Optional[TradBode]); der instalonde); while labe while labe node = wood.latt state = late[afs(s[rost)) while seath and tradb = "stath.opt());</pre>		
postorder traversal	<pre>def traversi[rost (ptimos[Transhode)); stat = rost and [rost, False)] station (state = rost (rost, False)) station (state = rost (rost), rost) state = rost (rost), (rost (rost), rost) state = rost (rost), (rost (rost), rost), (rost (rost), rost) yield sode (rost)</pre>	dat trevenskipent: "Demi-"; sheek = root and [Cont, Takes) whis statiste statiste statiste statiste statiste statiste (mot spain) tatist = ([mot, Trus)] + ([kid, Takes) for kid in mode,children[:-1]) till mode,val proid mode,val	def treversi(mif, graph, statt); state, sem = ((stat, fulsy), (statt) state, sem = (stat, fulsy); state, sem = (stat, fulsy); state = ((stat, fulsy), (statt); state = ((stat, fulsy), (statt), (statt); state = ((stat, fulsy), (statt), (statt); state = ((statt, fulsy), (statt)); state = ((statt, fulsy), (statt), (statt), (statt); state = ((statt, fulsy), (statt), (statt), (statt); state = ((statt, fulsy), (statt), (statt), (statt); state = ((statt, fulsy), (statt), (statt), (statt), (statt); state = ((statt, fulsy), (statt), (statt), (statt), (statt); state = ((statt, fulsy), (statt), (statt), (statt), (statt), (statt); state = ((statt, fulsy), (statt),
допускаются дубон точен в стене			def treversigraph); while stack: Mindow w mand.pop() DEC 0 mass Mindow W mandal DEC 0 mass Mindow W mandal DEC 0 mass Mindow Mindo
			mem robbs her for hid in reversed(graph/code)) 1000 return
разное			of dispersequent, startly 4 relatives upon the control of the cont
			of two princip graph last > latinil of procession for fact in preplaced [Elizan to mem] or are in seen - set[) stem littlettatio.chain.from_iterals[risem.adjoinds or notes
			can temp_sericolf, graph! (Panes sign inp of collections.connect() of the Ain graph) [tap-quadret((Mai: 1)) for mode in graph for kid in graph[mode]] for topo * (for ky k v in Lupture)] if v o * 0] [tap-quadret((kid: 1)) or inp[kid] * * C and taps_separe((kid) for mode series large f mode mode involved (mai: 1).
			def montreat, vieness ilmillateity]) >> limillate]; def = collections.metalicin(112); [qrid=1, speed)] for a, b, in include] empropress aspossopra votame def factors, person, real); white grid=106()]
			white gridphost; Thirdichost() put(), node, res) FRANCESCORD () put parameter as speed companies climateries and FRANCESCORD () put parameter as speed companies climateries and frances for put parameter () put rest[1-1] def formeds, pervi) print(print, put put print(print) print(print) put
			grid = list(map(collections.Counter, graph)) fo(), home) out or collections.counter, graph) out outer(main, ticker; list(tist(pty)) >> list[sty];

def unior(maif, tickeus idat[idat[ut]) > Limitus]
grid, stack, remult = olderinas/statisficitisty | ..., []
[grid[s].appen(b) for s, b in tickeus | topropyres arycompte tomace
while stack;
while stack = grid[code] = stack([1]);
while the control of the code of the code

	graph's, вместо всел словарь оставшимся вершин	graph land
n, sec m		max = [[0, 0, 0], [0, 1, 0], [1, 0, 0]] [def meliphoretest, grid, painty] for y, a in (general 2, pointy], quincy] for y, a in (general 2, pointy), [for y, a in (general 2, pointy], quincy] [for supportedly grid, pointy] for dy, do in (q-1, q-1, q-1, q-1, q-1, q-1), [1]; [for y, a in (q-1, q-1, q-1, q-1, q-1, q-1, q-1, q-1,
or seed of the General Mark (Mark 1888 and 1) and seed of the General Ma	def folded, empt) print(nede) print(nede) fold, calculations (contempty physical Director population along) folf, calculations (contempty physical Director property) of folded (contempty) of folded (contempty) enter the contempty physical	one fragminately [], ***********************************
h(mode) M. kid. on in seen) S in graph(mode) M. kid. on in seen) if level wise []	of traversi [saf, graph]: ivex, sty = [0]; culteriors.Counter[graph.baye] minis level: ivex = [tid for node in level for bid in graph[node] # stay_pup[bid, folion] def foliows,	def traversalignid, #EERCH(, D); level, same '[fast], ED; while level' level same '[fast], ED; while level' level same level same
FC_1C_38800	def trevered(self, graph) -> int: quese, stay = ordentions.depos(EE), ordentions.Counter(graph.keys()) blind	der followed: [0, 0), secondary [1] return level and for public level for hid in public level for hid
	out foliose, stay): extinct (set for kid in graph[outed] if stay) = (kid, best) for ear in folkid, stay)) or [oute) foll; collections. Content graph, beyon () def foliosit(): A gas measurements graph for seas, measure type a format extinct (loads): pask for als in graph[oute] for pask in folkid) or [(outed)]	
or hid in graph(node) N. hid out in week)	def treversiment, graph, started; stack, expy "(intar, Fales); ellections_Counter(graph.loge); stack, expy = (intar, Fales); inter(a = (inote, row)) * ((init, Fales) for hid in graph(node) f. stayogn(hid, Fales) stack = ((inote, row)) * ((init, Fales) for hid in graph(node) f. stayogn(hid, Fales) stack = (inote, row) stack =	
Company or are in to (Add, seem) The (Add, see	def traversilenif, graphi def traversilenif, graphi white start: if MENTALIMONO = starks,mpu(), falso; start = (lid for hid in reversedgraph(code)) MENTALIMONO def follows, server; if MENTALIMONO = (lid for hid in reversedgraph(code)) MENTALIMONO def follows, server; if MENTALIMONO = (lid for hid in reversedgraph(code)) MENTALIMONO = (lid for hid in reversedgraph(code)) if the code (lid follows) for hid in graphicode) MENTALIMONO = (lid follows, server) if the code (lid follows) for hid in reversedgraph(code)) MENTALIMONO = (lid follows, server) if the code (lid follows) for hid in reversedgraph(code)) MENTALIMONO = (lid follows, server) if the code (lid follows) for hid in reversedgraph(code)) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follows) code (lid follows) for hid in reversedgraph(code) MENTALIMONO = (lid follow	def they, will in the second of the set of the second of
milenday, these () [All and a maj	def dijektralmin, graph, mast): F origined graph with heap: it has been been been been been been been bee	
In (meen.mid(hid) or hid) + (mode)	def top, part(spelf, peph: list) > list[int]; def [choose] return (par for hid in prophrobe) return (par are in fo[hid]) * [mode]	
in topo for kid in graph[node]]		

KOHCHEKT Page 2

def transport(print, start([5, 10]))

seate - 60000

seate - 60000

if start([5, 10]) - 50000

seate - 50000

if start([5, 10]) - 50000

seate - 500000

seate - 50000

seate - 500000

seate - 50000

seate - 500000

seate - 50000

seate - 50000

seate - 50000

seate - 50000

s

gates = [(y, x) for y, row is enumerate[rows] for x, col is enumerate[row] if not col]
for gatey, pates in gates: f now cpany or rows as new wavege copous
[1] (gatey, gates = 1);
[1] (gatey = 1);
[1] (gatey = 1);
[2] (gatey = 1);
[3] (gatey = 1);
[4] (gatey = 1);
[5] (gatey = 1);
[6] (gatey = 1);
[7] (gatey = 1);
[8] (gatey = 1);
[9] (gatey = 1

def fo(y, s)

protein(y, s), falsel;

def g(y, s), false

 $\frac{\det f(x_j,x_j)}{\det g(x_j,x_j)} = \frac{\det (x_j,x_j)}{\det g(x_j,x_j)} = \frac{\det (x_j,x_j)}{\det g(x_j,x_j)} + \frac{\det (x_j,x_j)}{\det g(x_j)} + \frac{\det (x_j,x_j)}{\det g(x_j)} = \frac{\det (x_j,x_j)}{\det g(x_j)} + \frac{\det ($

def travermal(smif, grid, start=(5, 0));
stath, grid = [start], ([y, u]) val for y, row in enumerate(grid)
will grid, grid, property;
if grid grid, property;
print(y := point([]) x := point([])
start := (y + 1, u), (y - 1, u), (y, x + 1), (y, x - 1)

der unne find(self, edges list[list[int]] -> List[int]:

"End " end " en

def union find(self, edges):
 parent = ".join(neg(chr, range(10000)))
 for i, ji nedges:
 if parent[i] == parent[j]:
 return i, j
 range = parent[j]: replace(parent[i], parent[j])

m_(x, y) or x != y