

1 BibTeX

Vključevanje literature je lahko: potrebujete le dva ukaza: `\bibliography{file}` in `\bibliographystyle{file}`. Citirate kot običajno s `\cite`.

2 Pisanje algoritmov

Za pisanje algoritmov sta na voljo okolji `algorithm` in `algorithmic` iz paketov `algorithm` in `algorithmix`, ki sodelujeta podobno kot `table` in `tabular`. Algoritmi plavajo med tekstom, enako kot slike in tabele, nanje se lahko tudi sklicujemo, kot prikazano v izvorni kodi in v algoritmu 1. Sklicujemo se lahko tudi na pomembne vrstice, npr. na vrstico 19, ki predstavlja glavni del algoritma. Za primer pisanja algoritma se posvetujte s primerom v tem dokumentu, za bolj napredne primere uporabe, kot na primer razbijanje algoritma na več kosov, pa z (precej razumljivo) uradno dokumentacijo¹.

Algoritem 1 Opis, ki ima enako funkcionalnost kot opis pod sliko.

Vhod: Števili $n, m \in \mathbb{N}, n > m$.

Izhod: Decimalno število x , ki aproksimira rešitev enačbe $nx = m$.

```
1: function REŠI( $n, m$ )                                ▷ Vsi vhodni parametri morajo biti opisani.
2:    $a \leftarrow []$                                     ▷ Spremenljivka  $a$  naj postane prazna kopica.
3:   for  $i \leftarrow 1$  to  $n$  do
4:     if  $i \bmod 7 = 5$  then
5:       HEAPOP( $a$ )
6:     else if  $i < 5$  then
7:       HEAPPUSH( $a, \frac{i+12}{7} + \pi$ )                    ▷ Lahko uporabljamo matematiko.
8:     else
9:       HEAPPUSH( $a, i$ )
10:    end if
11:  end for
12:   $x \leftarrow 0$                                        ▷ Prazna vrstica
13:  for each  $e$  in  $a$  do                                ▷ To je primer komentarja.
14:     $x \leftarrow 1 + \sqrt[e]{x}$ 
15:  end for
16:  while  $|x| > \varepsilon$  do
17:     $x \leftarrow x/2$ 
18:  end while
19:   $x \leftarrow m/n$ 
20:  return  $x$                                           ▷ Vsi izhodni parametri morajo biti opisani nad algoritmom.
21: end function
```

3 Izvorna koda

Paket `minted`². Potrebuje program `pygmentize` za barvanje kode.

¹<http://tug.ctan.org/macros/latex/contrib/algorithmicx/algorithmicx.pdf>

²<http://ctan.ijs.si/tex-archive/macros/latex/contrib/minted/minted.pdf>

```

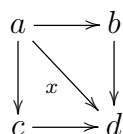
1  a = input()
2  for i in range(10):
3      print(i)

```

Lahko tudi iz datoteke ali inline.

4 Diagrami

Uporabljamo paket `xy`³. User guide: <http://ctan.ijs.si/tex-archive/macros/generic/diagrams/xy-pic/doc/xyguide.pdf>



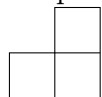
5 MakeIndex

Izdelava stvarnih kazal.

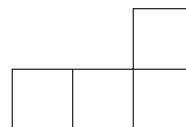
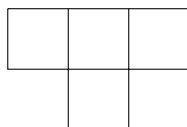
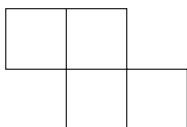
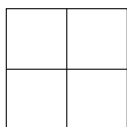
6 Tikz

Risanje risbic. Navodila: <https://www.bu.edu/math/files/2013/08/tikzpgfmanual.pdf>.

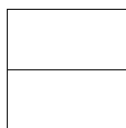
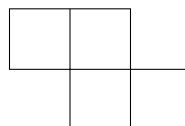
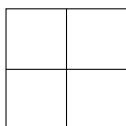
Preproste oblike; lahko kar tako, 



lahko pa v okolju za sliko.

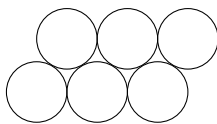


Bolj modra rešitev, če imamo več slik: vsako sliko damo v okolje "scope", vsaki podsliki lahko določimo `xshift` in `yshift`, na ta način lahko postavimo več slik.

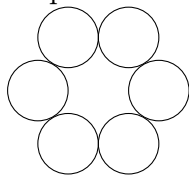


³<https://en.wikibooks.org/wiki/LaTeX/Xy-pic>

Še en primer s krogci. A:



B:



Najpomembnejše: kako s tikzom risati grafe.

Lahko si že takoj za `begindocument` definiramo stil za vozlišča, ni pa treba. Tole je ok, ampak je problematično, če želimo tudi kakšno oznako narisati kot vozlišče:

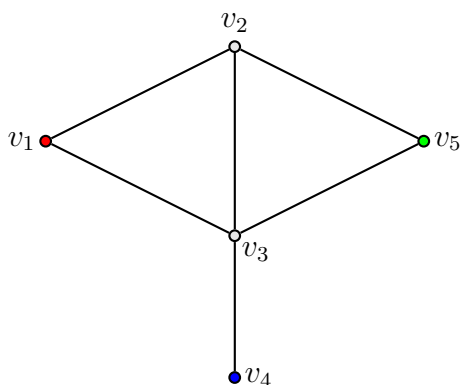
```
% Define style for nodes
\tikzstyle{every node}=[circle, draw, fill=black!10,
inner sep=0pt, minimum width=4pt]
```

Opomba: meni je všeč, če so vozlišča notri rahlo siva. To je stvar osebnega okusa. Mislim, da ima velika večina ljudi raje notri povsem belo, tako da je morada bolje, da generičen stil nastaviš tako. Da ne bo imel Klavžar težav, ker mu bodo vsi risali v tem stilu ;)

Bolje je nekaj takega:

```
\tikzstyle{noeud}=[circle,inner sep=2, minimum size =3 pt, line width = 1pt, draw=black]
```

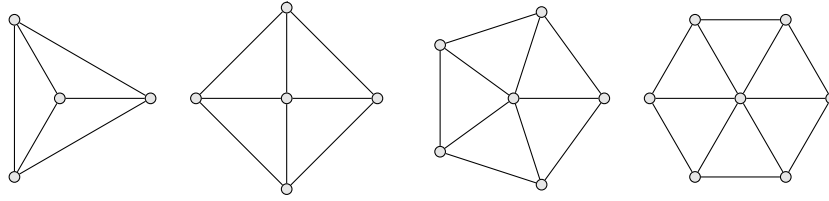
Lahko pa stil definiramo tudi pri vsaki sliki posebej (primer tega je spodaj).



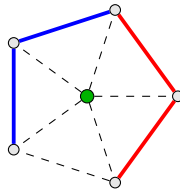
Slika 1: Graf G . Vozlišča lahko tudi obarvamo in označimo. Pri oznakah lahko sami določimo kje bo oznaka (levo, desno, oddaljenost, kot).

Mimogrede, povezavo med x in y lahko narišemo na več načinov: `\ path (x) edge (y);` ali `\ draw (x) edge (y);` ali `\ draw (x) -- (y);` (tu vmes sta dva minusa). Seveda lahko narišemo tudi ukrivljeno povezavo: `\ draw [bend left=20,-] (x) to (y);`

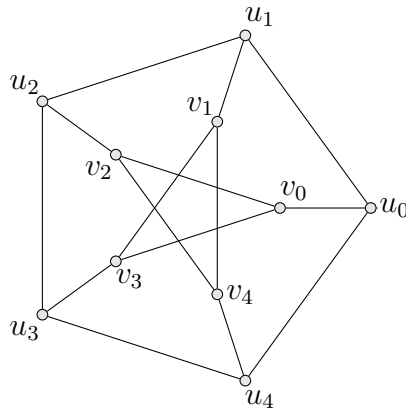
Prikazan še drug stil vozlišč (z nekoliko neposrečenim imenom, morda ga raje spremeni). Tu je stil definiran znotraj slike. Tudi stil za node je popravljen, sicer oznake niso lepo narisane (ampak so v sivih krogcih). Za tole je treba dodati še paketek `decorations.pathreplacing`, da imaš tiste lepe zavite zaklepaje zgoraj. Mogoče je ta primer že preveč zapleten in raje drugačen stil za vozlišča pokaži na enem od zgornjih lažjih primerov.



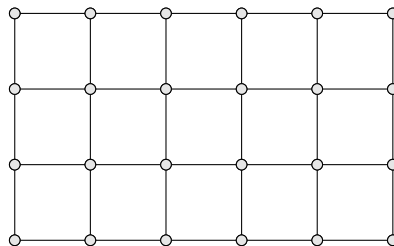
Slika 2: Grafi koles W_3, W_4, W_5 in W_6 . Koordinate točk lahko podamo tudi s polarnimi koordinatami - kulsko. Zelo uporabna zadeva je for zanka, ker je potem manj pisanja. Razložim več (predvsem zale lastx)?



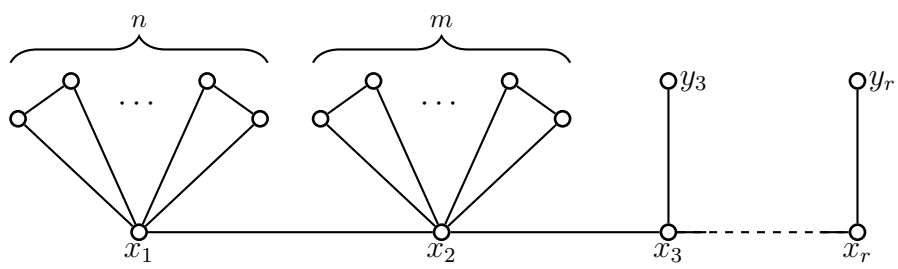
Slika 3: Tudi s črtami lahko delamo marsikaj.



Slika 4: Petersenov graf.



Slika 5: Graf $P_6 \square P_4$ (mreža).



Slika 6: En lep graf.