

# Random Access on Narrow Decision Diagrams in External Memory

---

**Steffan Christ Sølvesten**, Casper Moldrup Rysgaard, and Jaco van de Pol

SPIN 2024



# Adiar

*I/O-efficient Decision Diagrams*

---

[github.com/ssoelvsten/adiar](https://github.com/ssoelvsten/adiar)

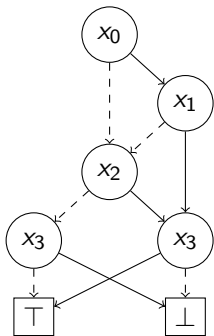
 Features

 Optimisations

 Features

 Optimisations





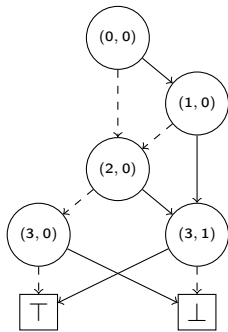
[ ((0, 0), (2, 0), (1, 0)) ,  
 ((1, 0), (2, 0), (3, 1)) ,  
 ((2, 0), (3, 0), (3, 1)) ,  
 ((3, 0), T, ⊥) ,  
 ((3, 1), ⊥, T) ]

**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Node-based representation of prior shown BDDs



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Priority Queue:  $Q_{count}$ :

[

]





**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Priority Queue:  $Q_{count}$ :

[

]

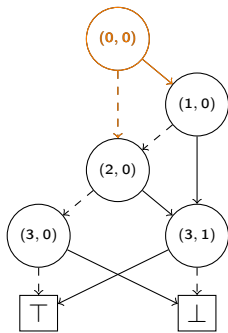


**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Priority Queue:  $Q_{count}$ :

[  $((0,0) \xrightarrow{\top} (1,0), 1)$  ,  
 $((0,0) \xrightarrow{\perp} (2,0), 1)$  ,

]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(1, 0)$	0	0

Priority Queue:  $Q_{count}$ :

[  $((0, 0) \xrightarrow{\top} (1, 0), 1)$  ,  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,

]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(1, 0)$	0	0

Priority Queue:  $Q_{count}$ :

[  $((0, 0) \xrightarrow{\top} (1, 0), 1)$  ,  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,

]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(1, 0)</b>	1	0

Priority Queue:  $Q_{count}$ :

[  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 ]

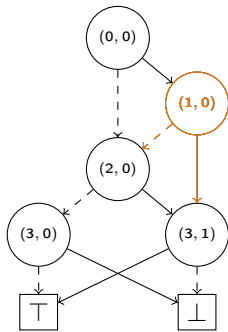


**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(1, 0)$	1	0

Priority Queue:  $Q_{count}$ :

[  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 $((1, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 ]



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(2, 0)$	0	0

Priority Queue:  $Q_{count}$ :

[  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 $((1, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 ]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(2, 0)</b>	0	0

Priority Queue:  $Q_{count}$ :

[

$((0, 0) \xrightarrow{\perp} (2, 0), 1)$	,
$((1, 0) \xrightarrow{\perp} (2, 0), 1)$	,
$((1, 0) \xrightarrow{\top} (3, 1), 1)$	,

]





**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(2, 0)</b>	1	0

Priority Queue:  $Q_{count}$ :

[  
 $((1, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 ]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(2, 0)$	2	0

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(2, 0)</b>	2	0

Priority Queue:  $Q_{count}$ :

[

$((2, 0) \xrightarrow{\perp} (3, 0), 2)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{\top} (3, 1), 2)$  ]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(3, 0)</b>	0	0

Priority Queue:  $Q_{count}$ :

[

$((2, 0) \xrightarrow{\perp} (3, 0),$	2	,
$((1, 0) \xrightarrow{\top} (3, 1),$	1	,
$((2, 0) \xrightarrow{\top} (3, 1),$	2	]



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(3, 0)	0	0

Priority Queue:  $Q_{count}$ :

[

$((2, 0) \xrightarrow{\perp} (3, 0), 2)$	,
$((1, 0) \xrightarrow{\top} (3, 1), 1)$	,
$((2, 0) \xrightarrow{\top} (3, 1), 2)$	]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(3, 0)</b>	2	0

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{T} (3, 1), 2)$  ]



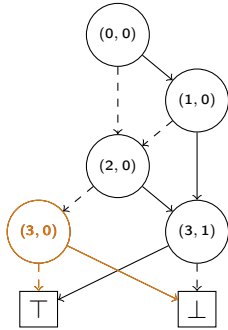
**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(3, 0)</b>	2	2

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1), \quad 1) \quad ,$   
 $((2, 0) \xrightarrow{T} (3, 1), \quad 2) \quad ]$



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(3, 1)</b>	0	2

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1),$	1)	,
$((2, 0) \xrightarrow{T} (3, 1),$	2)	]





**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(3, 1)</b>	0	2

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{T} (3, 1), 2)$  ]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(3, 1)$	1	2

Priority Queue:  $Q_{count}$ :

[

$((2, 0) \xrightarrow{\top} (3, 1), \quad 2) \quad ]$



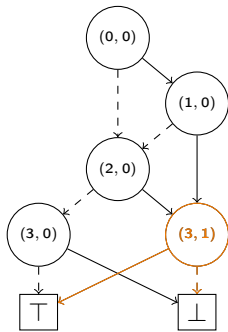
**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
<b>(3, 1)</b>	3	2

Priority Queue:  $Q_{count}$ :

[

]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek  
**(3, 1)**

Sum  
3

Result  
5

Priority Queue:  $Q_{count}$ :

[

]



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Result  
5

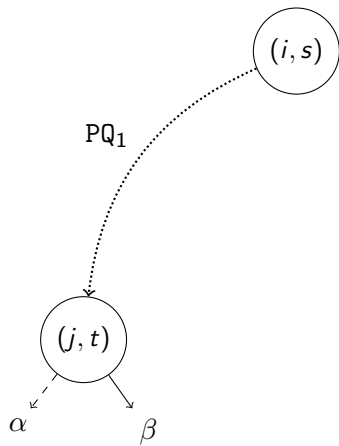
Priority Queue:  $Q_{count}$ :

[

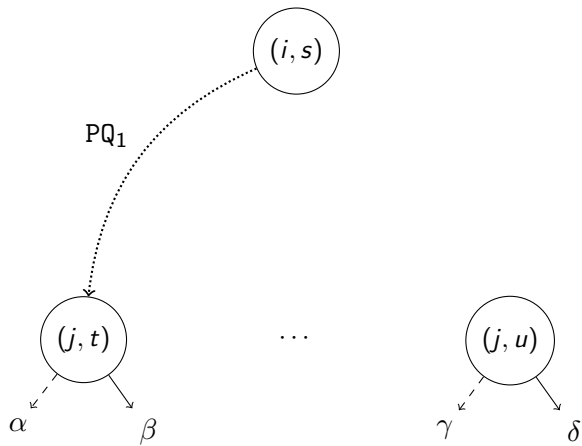
]

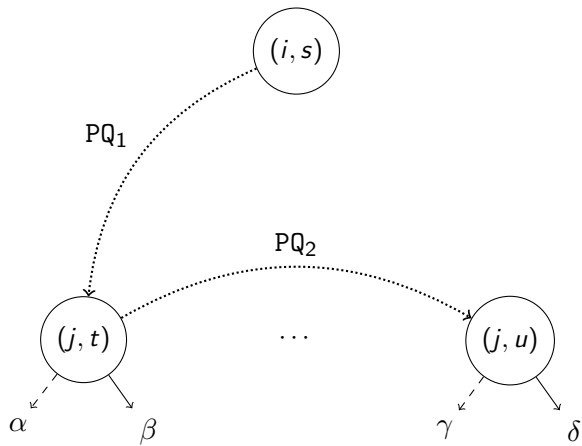


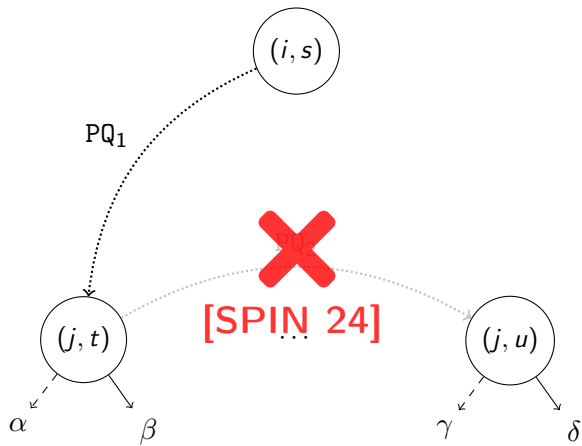
$$(i, s)$$

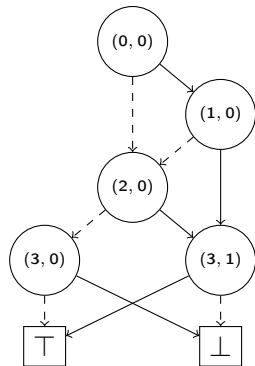


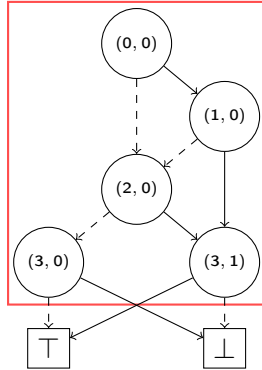


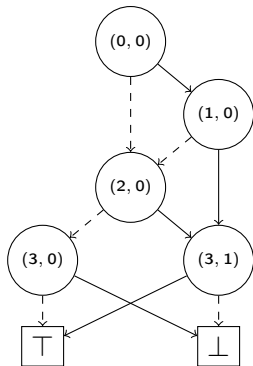


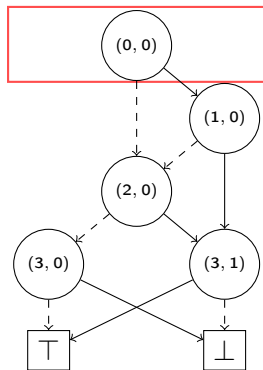


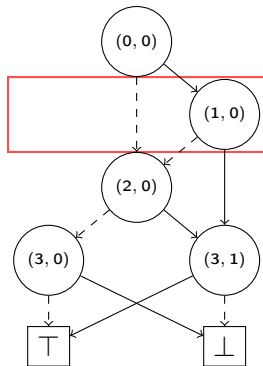




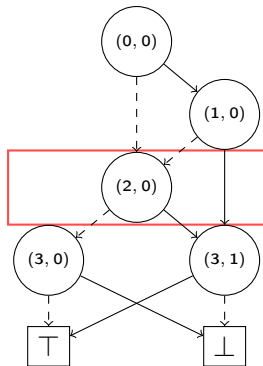


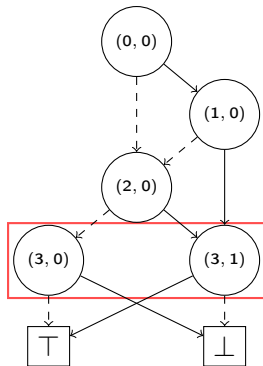


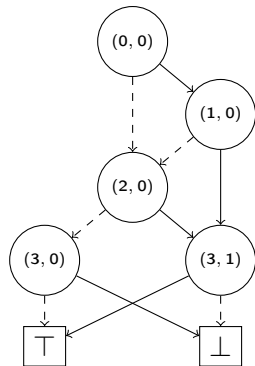


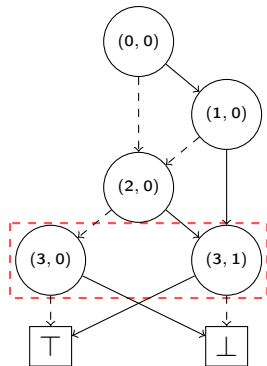








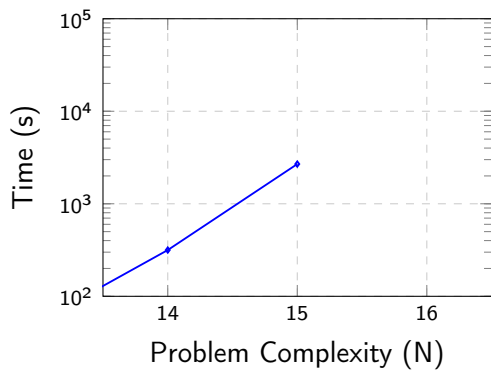






Width

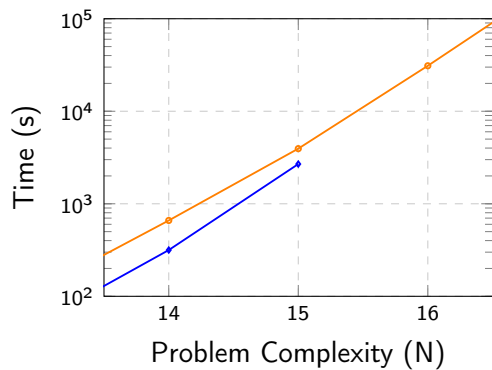
2






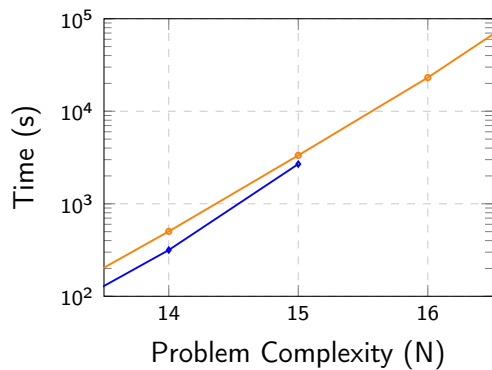
Queens | 300 GiB of RAM

		
		$N = 15$
	CUDD v3.0	: 44.8 min




Queens | 300 GiB of RAM

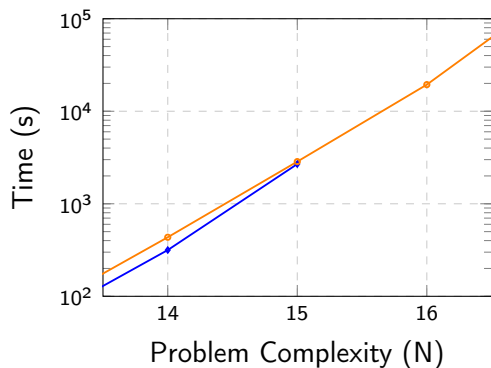
<div>  </div> <div><math>N = 15</math></div>			
◇	CUDD	v3.0	: 44.8 min
○	Adiar	v1.0	: 66.7 min




Queens | 300 GiB of RAM

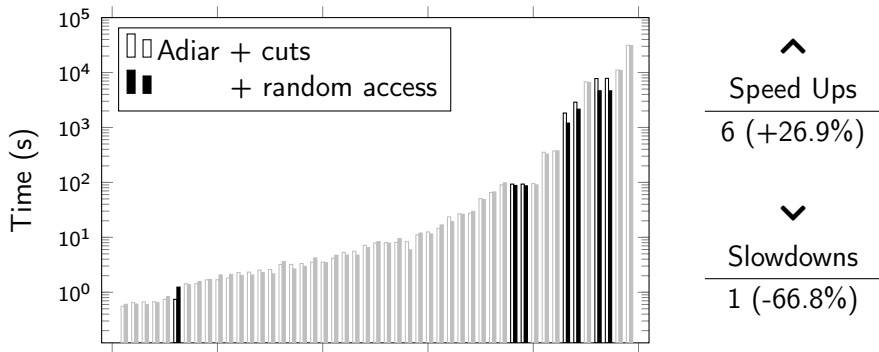
<div>  </div> <div><math>N = 15</math></div>			
◇	CUDD	v3.0	: 44.8 min
○	Adiar	v1.0	: 66.7 min
	+ cuts		: 56.8 min





Queens | 300 GiB of RAM

<div>  </div> <div><math>N = 15</math></div>			
◇	CUDD v3.0	:	44.8 min
○	Adiar v1.0	:	66.7 min
	+ cuts	:	56.8 min
	+ random access	:	47.2 min



EPFL Circuit Verification | 300 GiB of RAM



# Steffan Christ Sølvesten

---

✉ [soelvsten@cs.au.dk](mailto:soelvsten@cs.au.dk)

🌐 [ssoelvsten.github.io](https://ssoelvsten.github.io)

## Adiar

---

🔗 [github.com/ssoelvsten/adiar](https://github.com/ssoelvsten/adiar)

📖 [ssoelvsten.github.io/adiar](https://ssoelvsten.github.io/adiar)

